spiraltrain@gmail.com www.spiraltrain.nl Tel.: (0)30 737 0661 Fax: (0)30 737 1050

KvK nr. : 61618039 BTW nr. : NL854415798B01 IBAN : NL58RAB00156271540 BIC : RABONL2U



# **ADE100: Course Object Orientation**

Code: ADE100

Duration: 1 day

**Price:** € 499

# **Audience Object Orientation Course**

The course Object Orientation is intended for individuals who want to become familiar with the basic concepts of object-oriented system development.

# **Prerequisites Course Object Orientation**

To join this course is no specific skills or knowledge is required. General knowledge of system design is helpful to a proper understanding.

## **Realization Training Object Orientation**

The theory is treated on the basis of presentation slides. Demos and exercises are used to illustrate the theory. The course material is in English. The course times are from 9.30 up and to 16.30.

# **Certification Object Orientation**

Participants receive an official certificate Object Orientation after successful completion of the course.

# **Course Object Orientation**

The course Object Orientation covers the basic concepts of object orientation such as objects, classes, messaging, encapsulation, polymorphism and abstraction. After an





# **Object Orientation**





introduction explaining the path that led to object orientation, it is discussed what classes are, how objects can be instantiated from classes and how responsibilities for data storage and processing can be assigned to classes. Also well known object oriented concepts like encapsulation, inheritance, polymorphism, interfaces and abstract classes are explained and demonstrated. The relationships that may exist between classes like associations, aggregations and composition are discussed. Finally, attention is paid to the standard methods and techniques of object oriented system design and modeling with UML.

# Module 1 : Intro Object Orientation

Characteristics of Software Software Crisis Object Oriented Paradigm Object Orientation in Software Process Domain Analysis Requirements Gathering Use Case Analysis Use Case Diagrams Object Orientation in Software Design Objects as Domain Concepts Objects as Program Concepts Reusability Object Oriented Programming Paradigm Unstructured Programming Procedural Programming Object Oriented Programming

### Module 4 : Object Oriented Modeling

Object Oriented Modeling with UML UML Diagrams and Views Static Modeling Class Diagram Generalizations Avoid Unnecessary Generalizations Associations Identifying Associations Aggregation Composition Object Diagrams Associations versus Generalizations Interfaces Dynamic Modeling Interaction Diagrams Sequence Diagrams

### Module 2 : Classes and Objects

Abstraction in Object Orientation Procedural versus OO View Objects Classes Instance variables Methods and Operations Class or Instance? Identifying Classes Identifying Attributes Assign Responsibilities to Classes Identifying Operations Prototyping on paper CRC Cards Constructors Creating Objects Using Objects

### Module 3 : Object Oriented Concepts

**Object Orientated Concepts** Other Key Concepts Encapsulation Access Control **Class Fields and Methods** Inheritance Inheritance Hierarchy ls a rule Method Overloading Method Overriding Polymorphism Polymorphism Example Abstract Classes Interfaces Interface Implementation Dynamic Binding

# ADE200: Course UML Overview

#### Code: ADE200

Duration: 1 day

Price: € 499

# Audience UML Overview Course

The course UML Overview is intended for developers, designers, architects, managers and other interested persons who want to get an overview of the Unified Modeling Language (UML) standard for modeling systems.

#### Prerequisites Course UML Overview

Knowledge of and experience with system development and object orientation is beneficial to a good understanding but not strictly required.

# **Realization Training UML Overview**

The theory is treated on the basis of presentation slides and is interspersed with exercises. UML models are used as demonstration of the concepts. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification UML Overview**

Participants receive an official certificate UML Overview after successful completion of the course.

#### **Course UML Overview**

The course UML Overview gives insight into the UML language for modeling systems.

It discusses the UML syntax, symbols, diagrams and views. After an introduction to the UML specification and the meaning of UML as a meta language, the courses addresses Structural Modeling and the diagrams used like class and object diagrams, component and deployment diagrams. Also domain modeling and the modeling of possible relationships between classes are discussed like inheritance, associations, aggregations, compositions and dependencies. Next attention is paid to Use Case Modeling, the Use Case Diagram, the role of actors and the accurate description of the interaction steps. The next subject is Dynamic Behavior Modeling where interaction diagrams like sequence diagrams and collaboration diagrams are discussed. The modeling of system states using state chart diagrams in which concepts such as control and data flow and swim lanes are discussed. Finally a number of advanced concepts such as the UML modeling of packages and subsystems and the use of stereotypes, constraints and tagged values are part of the program.

Module 1 : UML Introduction	Module 2 : Use Case Modeling	Module 3 : Structural Modeling
What is UML? Modeling Reasons UML History UML Design Goal UML Diagrams UML Views Use Case View Logical View Component View Deployment View Notes and Adornments Stereotypes Tagged Values Contraints System Sequence Diagrams	Use Cases Actors System Context Diagram Identifying Use Cases Use Case Diagram Use Case Modeling Steps High Level Use Case Expanded Use Case Structuring Use Case Model Include Relationship Include Use Case Extends Relationship Extends Relationship Extends Use Case Use Case Generalization Actor Generalization	Structural Modeling Identification of Classes Structural Diagrams Elements Structural Relationships Association Modeling Domain Class Model Interfaces Composition Generalization Dependencies Packages Objects and Links Component Diagrams Deployment Diagrams Design Class Diagrams
Module 4 : Interactions	Module 5 : State Machines	Module 6 : Activity Graphs
Interaction Diagrams System Sequence Diagrams Object Sequence Diagrams UML Messages Sequence Diagrams Elements Recursion and Conditions Sequence Diagram Syntax Communication Diagrams Communication Diagram Elements Communication Diagram Syntax Interaction Diagrams are Valuable	State Machines State Diagram Elements State Machine Usage State Entry and Exit Actions Pseudo States Order of Actions Internal Transitions State Do Activities Guards History States State Diagram Syntax	Activity Diagram Steps in Activity Diagrams Actions and Subactivities Activity Diagram Elements Decisions Sync State Fork Transitions Join Transitions Swim Lanes Activity Diagram Syntax Using Activity Diagrams





# ADE300: Course Object Oriented Analysis and Design

Code: ADE300

#### Duration: 5 days

#### Price: € 2250

### Audience Object Oriented Analysis and Design Course

This course is designed for developers and architects who want to learn object oriented analysis and design techniques and UML to design systems.

#### Prerequisites Course Object Oriented Analysis and Design

To join this course knowledge of the basic principles of object orientation is required and experience in object oriented software development is desirable.

#### **Realization Training Object Oriented Analysis and Design**

The subject matter is treated on the basis of presentation slides. During the course two case studies are developed from requirements to design. A modern UML tool is used to draw UML diagrams in it. The course material is in English. The course times are from 9.30 up and to 16.30.

### **Certification Object Oriented Analysis and Design**

Participants receive an official certificate Object Oriented Analyis and Design after successful completion of the course.

#### **Course Object Oriented Analysis and Design**

In this course you will learn the object oriented ways of thinking and techniques to





# Object Oriented Analysis and Design





analyze, design and model a software system as a collection of cooperating objects. The UML language runs as a central thread through the course. After an introduction and review of the key object oriented concepts and principles, the modern system development principle of iterative and incremental development is discussed. Next attention is paid to how the requirements of a system can be analyzed and how the typical forms of system use can be described with uses cases. After an overview of UML, it is discussed how a domain model can be established, how the various objects can be distinguished together with their attributes and relationships, and what information they exchange. Attention is paid to how responsibilities can be assigned to objects and how these can be translated and made visible with interaction modeling using sequence and collaboration diagrams and state charts. The various patterns that can be used in this process are also discussed. Part of the subject matter is also how the translation of the analysis model to a design class model can take place, including the design of a logical architecture with packages and subsystems and the mapping to code. The course also considers aspects of architectural design that are dealt with using component and deployment diagrams. Finally the focus is on the importance of design patterns to implement standard solutions.

Module 1 : Software Process	Module 2 : Requirements Analysis	Module 3 : Use Case Modeling
Software Development Process Software Development Phases Good Software Characteristics Building Good Software Capability Maturity Model Iterative Development Incremental Development Requirements Capturing Requirements Analysis Process System Design Test Driven Development Waterfall Model Evolutionary Development Software Errors Unified Process UP Phases OO Analysis and Design	Understanding Requirements Vision Documents Requirement Analysis Activities Requirement Types Functional Requirements Non-Functional Requirements Requirements Determination Requirements Classification Requirements Specification Conflicting Requirements Requirements Risks The glossary	Use Cases Actors Use Case Modeling Identifying Actors System Context Diagram Identifying Use Cases Use Case Diagram Use Case Modeling Steps High Level Use Cases Expanded Use Case Use Case Template Branching with If Alternative Paths Scenarios Structuring Use Case Model Generalizations include and extends
Module 4 : UML Overview	Module 5 : Domain Modeling	Module 6 : Use Case Realization
What is UML? UML History UML Design Goals UML Diagrams UML Views Use Case View Logical View Component View Deployment View Notes and Adornments Stereotypes Tagged Values Contraints System Sequence Diagrams	Why Domain Modeling? Domain Model Conceptual Classes Noun Identification Physical and Conceptual Objects Filter Rules Similar Conceptual Classes Types of Classes Domain Analysis Classes UML Notation and Classes Lowering Representational Gap Finding Associations Roles in Associations Multiplicity and Associations Generalization and Specialization Aggregation and Composition Finding Attributes Association Classes	Realizing Requirements System Behavior Input Events and Operations Sequence of Events Event Trace Diagram System Sequence Diagrams Scenario with Events Derivation from Use Case Naming System Operations Recording System Operations Contract Sections Postconditions Responsibility Driven Design Class Responsibilites Class Collaborations Interaction Diagrams Sequence Diagrams Grasps Design Patterns
Module 7 : Statecharts	Module 8 : Design Model	Module 9 : Architectural Design

State Machine Concepts State Machine Diagram Diagram Elements State Machine Usage Event Driven Behavior State Machines and Objects **Object Behavior** Objects and Threads Passive Objects Active Objects Entry and Exit Actions Order of Actions Internal Transitions State Activities Guards History States

## Module 10 : Applying Design Patterns

What are Patterns? Creational Patterns Behavioral Patterns Structural Patterns Architectural Patterns Singleton Abstract Factory Factory Method Reducing Dependencies Observer Pattern Adapter Pattern Façade pattern Proxy Pattern Transition to Design From Requirements to Design Class Design Diagrams The Design Model Design Aspects Design Model Characteristics Mapping to Code Packages Package Design Packaging Guidelines Data Access Class Packages Subsystems System Partitioning Large Scale Element Collaboration Consider Layering Layers and Packages Simple Logical Architecture Consider Coordination Layer Web Application Architecture Consider MVC Architecture Package Dependencies Ordering Work Load Test Early Clustering Vertical Scaling Horizontal Scaling Physical Architecture

# **ADE400: Course Design Patterns**

Code: ADE400

Duration: 3 days

### **Audience Design Patterns Course**

The course Design Patterns is intended for experienced developers and software architects with knowledge of object oriented programming and systems analysis who want to apply Design Patterns when designing these systems.

#### **Prerequisites Course Design Patterns**

Knowledge of an object-oriented programming language like C++, C#, or Java and experience with object oriented analysis and design with UML is required.

#### **Realization Training Design Patterns**

The concepts are treated according to presentation slides. The theory is illustrated with demos of patterns in C++, C# and Java. There are exercises in design problems where patterns are applied. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Design Patterns**

Participants receive an official certificate Design Patterns after successful completion of the course.

#### **Course Design Patterns**





# **Design Patterns**





In the course Design Patterns you will learn how design patterns can be applied to the object oriented design of systems. After an introduction about the role that design patterns play and how they can be used to realize the non-functional requirements of systems, attention is paid to how design patterns are described and cataloged. Also the role of design patterns in the architecture of applications is discussed and the various categories of design patterns that are distinguished. In the module Creational Patterns the Factory patterns and the Builder, Prototype and Singleton pattern are discussed. You learn out of which classes, relationships, responsibilities and cooperations a typical design pattern solution can consist. Next in the module the Structural Patterns the Adapter, Composite, Bridge, Decorator, Proxy and Flyweight pattern are discussed. You will learn the consequences of applying the patterns, the benefits and possible disadvantages in terms of time and space considerations and how to decide on the use of a particular pattern. Next in the module Behavioral Patterns the Chain of Responsibility, Interpreter, Iterator, Mediator, State and Strategy patterns are discussed. Finally the module Architectural Patterns considers certain patterns that are involved in the architectural structure of software including Operating Systems and Frameworks. This module focuses on the Layer pattern, the Micro Kernel pattern and the Model View Controller (MVC) pattern.

Module 1 : Intro Design Patterns	Module 2 : Creational Patterns	Module 3 : Structural Patterns
What is a design pattern? Describing design patterns Reuse through design patterns Structure of patterns Classification of patterns Catalog of Design Patterns Creational Patterns Structural Patterns Behavioral Patterns Sample design patterns Selecting Design Patterns Solving problems with design patterns	Factory Patterns Factory Method Pattern Connect parallel class hierarchies Abstract Factory Pattern Concrete Class Isolation Promoting Consistency Builder Pattern Controlling the build process Prototype Dynamic configuration Singleton Pattern Controlled access	Adapter Pattern Pluggable Adapters Composite Pattern Sharing Components Decorator Pattern Lots of little objects FaÇade Pattern Reducing client-subsystem coupling Flyweight Pattern Reducing number of instances Proxy Pattern Copy-on-write
Module 4 : Behavioral Patterns	Module 5 : Architectural Patterns	
Chain of responsibility Command Pattern Interpreter Pattern Iterator Pattern Mediator Pattern Memento Pattern Observer Pattern State Pattern Strategy Pattern Template Pattern	Architectural patterns versus design patterns Patterns for real-time software Layers Pipes and Filters Blackboard Broker Model-View-Controller Presentation-Abstraction-Control Microkernel Reflection	

# ADE500: Course Model Driven Architecture

Code: ADE500

Duration: 1 day

### **Audience Model Driven Architecture Course**

This course is designed for experienced developers and architects who want to learn about Model Driven Development and Model Driven Architecture (MDA) and use it in the development and design of applications.

#### **Prerequisites Course Model Driven Architecture**

To join this course knowledge and experience with system development and object oriented principles and UML is required.

#### **Realization Training Model Driven Architecture**

The course has a hands-on nature. The theory is treated on the basis of presentation slides and the theory is interspersed with demos and exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Model Driven Architecture**

Participants receive an official certificate Model Driven Architecture after successful completion of the course.

#### **Course Model Driven Architecture**

In this course you will learn the principles behind Model Driven Development and

Model Driven Architecture (MDA). After an introduction to the various Model and Metadata specifications of the OMG, the advantages of using MDA is discussed and the main features of DSLs, Domain Specific Languages, are addressed. Attention is paid to the MDA specifications and technologies that enable and support MDA such as UML, Object Constraint Language (OCL) and XMI. Also meta modeling and the Meta Object Facility (MOF) is a subject in the course. Further attention is paid to the MDA process, the role of the Platform Independent Model and Platform Dependent Model. In this respect the support for Model Transformations and code generation in different tools is discussed. Implementation of parts of MDA in the Eclipse Modeling Framework is part of the subject the subject matter. Both UML diagrams, Java interfaces and XML Schema are used as model input. Finally, the Microsoft approach of Model Driven Development is discussed as demonstrated in the Oslo Modeling project or SQL Server Modeling. The several M-languages are explained and demonstrated.

Module 1 : MDA Introduction	Module 2 : OMG Standards	Module 3 : MDA
What is Modeling? Models are Everywhere Object Management Group (OMG) Who are OMG? OMG Model and Metadata Specifications CORBA What is MDA MDA Characteristics Kernel Idea of MDA MDA Directions Standardized Meta Models MDA Modeling MDA Information and Tools	Unified Modeling Language (UML) UML Diagrams UML Extensibility Elements Stereotypes Tagged Values Constraints OCL Meta Object Facility (MOF) MOF Layers MOF Automated Translations UML Metadata Specifications XML Meta Interchange (XMI) From UML to XML	MDA Terminology MDA Helicopter View Platform Models Basic Principles of MDA Building an MDA Application Platform Specific Model Marking a Model Multiple Middleware Models Model Transformations Generate Implementation
Module 4 : EMF	Module 5 : Oslo	
What is EMF? EMF versus MDA What is an EMF Model EMF Model as Java Interface EMF Model as UML Diagram EML Model as XML-Schema EMF Architecture EMF Components EMF Generation Change Notification Reflection	What is Oslo? Key Oslo Concepts Oslo's Strategy Oslo Architecture M Language Semantic Model in Mschema MSchema Generated SQL MGrammar Song Grammar MGrammar and MSchema MGraph	







MDA

# ADE600: Course Service Oriented Architecture

Code: ADE600

Duration: 2 days

# Audience Service Oriented Architecture Course

This course is intended for experienced developers and architects who want to apply Service Orientation in developing and designing applications.

#### **Prerequisites Course Service Oriented Architecture**

To join this course knowledge and experience in system development and object oriented principles required.

#### **Realization Training Service Oriented Architecture**

The course has a hands-on nature. The theory is discussed on the basis of presentation slides and is interspersed with practical exercises. The course times are from 9.30 up and to 16.30.

# **Certification Service Oriented Architecture**

Participants receive an official certificate Service Oriented Architecture after successful completion of the course.

#### **Course Service Oriented Architecture**

In this course the principles and standards of a reusable service-oriented architecture (SOA) are treated. You will learn how candidate services can be deduced from

business processes. Attention is paid to service composition and decomposition and you will learn how to orchestrate services and processes using the Business Process Execution Language (BPEL). Further the standards which you can use in an SOA environment for creating interoperable services like SOAP, WSDL and XML Schema, are discussed. You will learn the importance of a contract first, WSDL first, approach in the design of services with a low degree of coupling. Finally attention is paid to the new WS standards that play a role in a SOA environment.

Module 1 : Overview SOA	Module 2 : Describing Services	Module 3 : Service Composition
Making the case for SOA Evaluating the benefits or services Surveying enabling technologies XML Schema, SOAP and WSDL Dissecting Services and Architecture Developing Services SOA architecture Service characteristics	Describing Web Services with WSDL Conveying information with SOAP / XML Characterization message entities Using schemas Composing a service description Defining the abstract interface Specifying communication protocols Service Modelling and Implementation	Structuring business requirements Breaking down business processes Identifying reusable operations Orchestrating Services with BPEL Automating service interaction with BPEL Executing the BPEL script Developing BPEL services Correlating Service Invocations

Module 4 : WS Standards

WS-I interoperability profiles Creating secure and reliable services WS-Security Global Web Service Architecture WS-Routing Other WS-Standards





# Service Oriented Architecture





# ADE700: Course Sparx Systems Enterprise Architect

Code: ADE700

Duration: 2 days

ENTERPRISE

**Enterprise Architect** 

Introduction

### Audience Sparx Systems Enterprise Architect Course

Software engineers, software architects and other future users of Enterprise Architect.

## Prerequisites Course Sparx Systems Enterprise Architect

To join this course basic knowledge of UML and object orientation is desirable.

### **Realization Training Sparx Systems Enterprise Architect**

The theory is discussed by means of presentation slides. The concepts are illustrated with demos and there is opportunity to practice. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Sparx Systems Enterprise Architect**

Participants receive an official certificate Sparx Systems Enterprise Architect after successful completion of the course.

#### **Course Sparx Systems Enterprise Architect**

The "Enterprise Architect" Training is a classroom training that is ideally suited for students with UML experience and experience with object oriented development to become acquainted with the UML modeling tool Enterprise Architect from Sparx Systems. The course follows the development cycle of an application. The course

provides students with the knowledge and the hands-on experience to make UML models and execute code generation using Enterprise Architect. After completion of the course, the student has an overview of the functionality of Enterprise Architect. The student has explored existing models and has knowledge of modeling aspects such as Requirements, Use Case Diagrams, Class Diagrams, Activity Diagrams, State Diagrams and Sequence Diagrams in Enterprise Architect. The student will gain broad knowledge about generating documents and web pages, about code generation and reverse engineering and about team collaboration and data modeling in Enterprise Architect.

Module 1 : EA Intro	Module 2 : Requirements and Use Cases	Module 3 : Domain Modeling
What is Enterprise Architect? UML Modeling Tool of Choice Share Models Capture Requirements Generate Documentation Code Generation Reverse Engineering MDA Transformations Model Databases Link EA to IDE's MDG Technologies	Understanding Requirements Vision Documents Requirement Types Functional Requirements Non-Functional Requirements Requirements Determination Requirements Classification Use Cases and Actors Use Case Modeling System Context Diagram Identifying Use Cases Use Case Diagram Scenarios Structuring Use Case Model include and extends	Domain Modeling Conceptual Classes UML Notation and Classes Associations Roles in Associations Multiplicity Naming Associations Generalization Specialization Aggregation Composition Attributes
Module 4 : Interaction Diagrams	Module 5 : MDG Technologies	Module 6 : MDA Transformations
Realizing Requirements From Analysis to Design Object Sequence Diagrams Responsibilities and Methods Class Responsibilities Class Collaborations Interaction Modeling Collaboration Diagrams Translate System Operations Diagram Notations Sequence Diagrams	MDG Products MDG Technologies MDG Examples Visual Studio.NET Eclipse Loading MDG Examples BPMN SysML	What is MDA? MDA Characteristics Kernel Idea of MDA MDA Directions MDA Terminology MDA Helicopter View Platform Models Basic Principles of MDA Building MDA Applications Platform Specific Model Marking a Model Multiple Middleware Models Model Transformations Generate Implementation



#### Price: € 1199

# **ADE800: Course Architectural Design Patterns**

Code: ADE800

Duration: 2 days

### Audience Architectural Design Patterns Course

The course Architectural Design Patterns is aimed at experienced developers and software architects who want to apply design patterns for the architecture of systems.

#### Prerequisites Course Architectural Design Patterns

Knowledge of an object-oriented programming language like C++, C# or Java and experience with object oriented analysis and design with UML is desired.

#### **Realization Training Architectural Design Patterns**

This course has a hands-on character. The theory is covered on the basis of presentation slides and is illustrated with demos of architectural patterns. There are exercises in design problems where architectural patterns can be applied. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Architectural Design Patterns**

Participants receive an official certificate Architectural Design Patterns after successful completion of the course.

#### **Course Architectural Design Patterns**

The course Architectural Design Patterns will discuss the importance and principles of

Architectural Modeling and the application of Architectural Design Patterns. Architectural Design patterns refer to the large scale organization of systems and the cooperation of their components and layers and not to the internal architecture of the individual software components. After a review of the basics of software design patterns and their terminology, attention is paid to the role of design patterns in the architecture of software systems. The major architectural pattern categories are discussed. Attention is for instance paid to Call en Return Patterns like the Layers pattern, to Data Flow Patterns like the Pipes and Filter pattern, to Shared Memory patterns like the Blackboard pattern and to Distributed Systems patterns like the Proxy and Broker patterns. For each pattern the benefits and drawbacks are discussed. Students will exercise and discuss the application of patterns for practical problems in software architecture. Next advanced patterns that involve concurrency and multiple threads are discussed and finally some patterns that do not belong to a specific category are discussed.

Module 1 : Software Architecture	Module 2 : Architectural Patterns	Module 3 : Call and Return Patterns
What is Architecture? Software Architecture Layered Architecture Software Characteristics Analogy to Building Role of Architect Software Architecture Elements Architecture Context Architecture Context Architectural Viewpoints Logical Architecture Non Functional Requirements Physical Architecture Early Load Testing	Architectural Modeling Model Characteristics Architectural Viewpoints Pattern Terminology Gang of Four Design Patterns Architectural Patterns Architectural Pattern Categories Batch Sequential Pattern Pipe and Filter Pattern Blackboard Pattern Publish and Subscribe Pattern Peer to Peer Pattern Model View Controller Pattern	Object Oriented Pattern OO Benefits and Drawbacks Object Oriented Architecture Layers Pattern Layers Problem Layers Solution Network Architecture Layers Benefits and Drawbacks Layers Pattern Variant Client Server Pattern Client Server Architecture Three Tier Pattern Three Tier Architecture
Module 4 : Data Flow Patterns	Module 5 : Shared Memory Patterns	Module 6 : Distributed Systems
Data Flow Architecture Batch Sequential Pattern Data Flow Pattern Problems Batch Sequential Pipes and Filter Pattern Pipes and Filter Forces Pipes and Filter Patterns Servlet Filters Web Service Handlers Call Chain of Handlers Benefits and Drawbacks Pipes and Filter Variants	Data Centered View Shared Repository Pattern Shared Repository Architecture Active Repository Pattern BlackBoard Pattern BlackBoard Architecture BlackBoard Context Speech Recognition BlackBoard Solution BlackBoard Variants BlackBoard Know Uses Benefits and Drawbacks	Proxy Pattern Types of Proxies Copy on Write Proxy Remote Proxy RMI Proxy Architecture Broker Pattern Broker Pattern Broker Solution Bridge Component Broker Variations Benefits and Drawbacks Broker Class Diagram
Module 7 : Interactive Systems	Module 8 : Implicit Invocation	Module 9 : Concurrency Patterns
MVC Pattern MVC Architecture MVC Model MVC View MVC Controller Multiple Views Same Data Known Uses MVC Benefits and Drawbacks PAC Pattern PAC Structure PAC Solution PAC Benefits and Drawbacks	Communication Patterns RPC Pattern Publish Subscribe Pattern Queue versus Publish Subscribe Topics and Queues Data Distribution Pattern Request Reply Pattern Request Reply Correlation Multiple Replies Scalable Request Reply Guaranteed Delivery Scenario Guaranteed Delivery	Reactor Pattern Server Socket Loop Reactor Pattern Forces Event Driven Design Single Threaded Reactor Non Blocking IO Thread Pool Reactor Reactor Known Uses Reactor Benefits and Drawbacks Active Object Pattern Active Object Method Scheduling Active Object Method Dispatch
Module 10 : Other Patterns		





# Architectural Design Patterns





MicroKernel Pattern Microkernel Structure State Machine Pattern State Machine Structure Reflection Pattern Reflection Structure Process Control Pattern Process Control Structure Master and Slave Pattern Master and Slave Structure

# **ADE900: Course Microservices Architecture**

Code: ADE900

### Duration: 2 days

Module 1 : Intro Microservices	Module 2 : Microservice Architecture	Module 3 : Service Communication
What are Microservices? Components and Services Loose coupling Passing Messages System Architectures Client Server and Multi Tier Monoliths and Maintenance Costs Splitting Into Modules Small Services Service Communication HTTP Resource API Centralized Management	Domain Driven Design Bounded Context Microservice Contract Single Source of Truth Ownership Independent Deployment Versioning Chattiness Dependency Cycles Sharing Code Duplication	Communication Patterns HTTP versus SOAP Service Contracts and WSDL Using an API Gateway REST and Resource ID's Stateless Communications Uniform Interface Standard HTTP Methods Content Negotiation Open API Specification GraphQL and gRPC Apache Thrift andAvro
Module 4 : Message Passing	Module 5 : Implementing Microservices	Module 6 : Discovery Patterns
WebSockets Server sent Events Message Queues Message Brokers JMS Apache Kafka Actor Model Akka Message Encoding and Decoding	Synchronous Model Sequential Execution Asynchronous Model Parallelism and Concurrency Handling Partial Failures Completable Future Thread Resources Blocking Operations	Client Side Discovery Load Balancing Service Registry Netflix Eureka Example Client Side Drawbacks Server Side Discovery Request Routing Kubernetes Anache Zookeeper

# Module 7 : Deployment Strategy

Multiple Services Pattern Process or Process Group Multiple Service Instances per Host No isolation drawback Service Instance Per Host Service Instance per Virtual Machine Tools for VM's Resource Utilization Drawback Service Instance per Container Pattern Docker and Kubernetes Serverless Deployment

# **ADM100: Course Apache Administration**

Code: ADM100

#### Duration: 3 days

#### Audience Apache Administration Course

The course Apache Administration is intended for system administrators or developers on Unix, Linux or Windows who should learn how to administer and control the Apache Web Server version 2.2 till 2.4.

#### **Prerequisites Course Apache Administration**

Knowledge of modern operating systems like Unix, Linux or Windows, in particular knowing how to deal with the file system, is desirable.

#### **Realization Training Apache Administration**

The theory is covered using presentation slides. The concepts are further explained using demos. The theory is alternated with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Apache Administration**

Participants receive an official certificate Apache Administration after successful completion of the course.

#### **Course Apache Administration**

In the course Apache Administration you will learn the workings of the Apache Web





# Apache Administration





Server versions 2.2 till 2.4 and how to install and configure Web Sites on Apache. You will learn the structure of the central configuration file, httpd.conf and the most important directives in it. Next attention is payed to the architecture of Apache, in particular the way modules, filters and handlers cooperate. Virtual Hosting is discussed and the different varieties of it that exist. Also logging is treated in particular the handling and structure of log files. Web Application Security is covered with HTTP Authentication and the configuration of SSL. And also securing the Apache Web Server itself and the use of mod\_security to protect against common attacks like Cross Site Scripting and SQL Injection is on the course schedule. Attention is further payed to the way filters and handlers hook into the request processing cyclus and how they can be configured. And also the manipulation of URL's with the powerful mod\_rewrite module is covered. Other topics are the configuration of a Proxy Server with mod\_proxy and how to integrate with the Tomcat Web Server for dynamic Java Web Applications. The measurement and improvement of the performance of applications through techniques like load balancing and caching is part of the course program as well. In the course attention is payed to both Apache under Linux as under Windows.

Module 1 : Intro Apache	Module 2 : Apache Configuration	Module 3 : Apache Architecture
Apache Web Server What is the ASF? Apache Name and Marketshare Apache Installation Apache Directory Structure Apache Directories Starting Apache Uniform Resource Locators (URL's) Domain Name Service (DNS) Static HTML pages Styling with CSS Dynamic Content Generation	Apache Configuration Files Server Configuration Directives Core Configuration Directives Authentication Directives Authorization Directives Three Pass Access Control Logging Directives Performance Related Directives Directive Containers Conditional Containers .htaccess Files AllowOverride Directive	Apache Architecture Overview Two Phase Operation Apache Portable Runtime Multi Processing Module (MPM) Modules Important Apache Modules Two Phase Operation Content Generation Request Processing Phases Processing Hooks Apache Filters Apache Handlers
Module 4 : Virtual Hosting	Module 5 : Logging	Module 6 : Web Application Security
Processing of Configuration Files Configuration Contexts Virtual Hosting Virtual Host Directives Name-based Virtual Hosts Host File Name-based Virtual Hosting IP-based Virtual Hosts Host File IP-based Virtual Hosting Multiple IP Adresses per NIC Port-based Virtual Hosts	Log File Types Error Logs Log Levels Access Logs Common Log Format Defining Custom Logs Log File Rotation Piped logs Log File Analyzers Server Log Reporting	HTTP Basic Authentication Configuring Authentication Managing Access Files Creating Passwords Secure Sockets Layer (SSL) SSL Encryption Types Asymmetric Encryption Symmetric Encryption Hash Encryption SSL Certificates
Module 7 : Hardening Apache	Module 8 : Mod Security	Module 9 : Handlers and Filters
Update Apache Enable Logging HTTP Request Methods Using SSL Set HTTP Limits Change Default User Block Directory Access Disable Directory Listing Hide Server Details Hide Etag Header Disable .htaccess Override Web Application Firewall Install mod_evasive	Firewall Limitations Common Exploits Protection Measures Intrusion Detection Systems Web Application Firewalls Evasion Issues What is ModSecurity? ModSecurity Features Rule Examples Rule Examples Rule Syntax Core Rule Set Default and Chained Actions Regular Expressions	What are Filters? Standard Apache Filters Simple Filtering Dynamic Filtering What are Apache Handlers? Built-in Handlers Server-Info Server-Status Why mod_rewrite ? Rewrite Flags Flow Control Rewriting versus Redirection Transforming URLs Conditional Rewrites
Module 10 : Tomcat and Apache	Module 11 : Web Proxy with mod_proxy	Module 12 : Performance and Scalability

What is Tomcat? Tomcat Binary Distributions Tomcat Zip Installation Tomcat Directories Server Configuration Files Other Configuration Files Tomcat webapp's Directory Tomcat Apache Communication Tomcat Workers Worker Properties Configuring AJP Mounting Tomcat Sites Forward Proxy Reverse Proxy / Gateway mod\_proxy Support Proxy Modules And Other Related Modules Configuring a Forward Proxy Storage of Cached Pages Configuring a Reverse Proxy Basic Reverse Proxy Config mod\_proxy\_balancer Proxying by Rewriting Backend Optimisations Access to the Proxy Server Performance Considerations Server Monitoring Web Server Log Files Logging and Performance Load Testing Operating System Tuning SSL Performance Caching Dynamic Content Vertical Scaling Horizontal Scaling Load Balancing Using mod\_proxy

# ADM200: Course Tomcat Administration

Code: ADM200

Duration: 3 days

### Audience Tomcat Administration Course

The course Tomcat Administration is intended for System Administrators and Web Developers who need to administer and control the Tomcat Server and who need to deploy applications onto it.

#### **Prerequisites Course Tomcat Administration**

Participants should be familiar basic computing skills like browsing the Web and accessing the directory structure. Knowledge of Web Applications and other Web Servers is beneficial.

### **Realization Training Tomcat Administration**

The theory is covered using presentation slides. The concepts are further explained using demos. The theory is alternated with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Tomcat Administration**

Participants receive an official certificate Tomcat Administration after successful completion of the course.

#### **Course Tomcat Administration**



# **Tomcat Administration**





The course Tomcat Administration covers the administration of the Apache Tomcat web server which is the most popular platform for deploying Java-based Web applications. The course starts with discussing the different installation options. Next it is shown how Java web applications using servlets and JSP's can be deployed on the server. The web application structure and configuration options with the deployment descriptor are also covered. Participants will become familiar with the internal architecture of the server, with JMX (Java Management Extensions) and the use of JMX to manage and monitor the server. Attention is also payed to the setup of virtual hosting and the different ways to secure web applications using authentication and SSL. It is further discussed how to integrate with the Apache Web Server which may serve the static pages or which may have the role of loadbalancer. The way Tomcat can be configured to enable Web applications to connect to databases is also a course subject. Finally it is discussed how to configure a cluster to ensure the failover in cases of server crashes and to enable the scalability of applications. The JMeter tool will be used to test the performance of Web applications.

Module 1 : Tomcat Introduction	Module 2 : Java Web Applications	Module 3 : Tomcat Architecture
What is Tomcat?	Servlets and JSP's	Structure server.xml
What is the ASF?	Servlet Characteristics	The Server
Apache Name and Marketshare	JSP Translation Time	The Service
Java Overview	JSP Request Time	Connectors
Java EE Servers	Form Submissions	Deployment Scenario's
Tomcat and JDK versions	POST and GET Data	The Engine
Servlet and JSP versions	Sessions	The Host
Tomcat Binary Distributions	Web Application Structure	The Context
Zip versus Exe Installation	WAR Files	Resources
Tomcat Directories	Deployment Descriptor	The Realm
Server Configuration Files	Defining Custom URL's	The Valves
Other Configuration Files	Preloading pages	Lifecycle Listeners
Webapps directory	Error pages	Apache Portable Runtime
Module 4 : Class Loading	Module 5 : Tomcat and JMX	Module 6 : Virtual Hosting
Class Loading Process	What is JMX?	Virtual Hosting
Class Loaders in JVM	JMX API	Name-based Virtual Hosts
Delegation Model	JMX Goal	Virtual Host Configuration
Custom Class Loaders	Where is JMX used?	Engine with Virtual Hosts
Class Loader Behavior	Managed Beans	Directory Structure Virtual Hosts
Class Loader Namespace	Standard Mbeans	Virtual Host Element
Custom Class Loaders	MBean Server	Host File Name-based Hosting
Tomcat Class Loaders	Naming MBeans	IP-based Virtual Hosts
System Class Loader	JMX Architecture	Multiple IP addresses per NIC
Common Class Loader	JVM Instrumentation MBeans	Separate JVM for Each Host
Web Application Class Loader	Accessing the JMX Agent	Server Configuration more JVM's
Class Loader Order	JMX in Tomcat	Host Configuration more JVM's
Module 7 : Connecting to Databases	Module 8 : Tomcat Security	Module 9 : Logging
Java Database Connectivity	Verifying Download Integrity	Logging in Java
JDBC Overall Architecture	Remove Default Applications	Java Util Logging
JDBC Executing a Statement	Change SHUTDOWN command	Levels and Log Methods
ClassNotFoundException	Special Tomcat Account	Tomcat Logging
Evolving JDBC versions	Securing JVM	Logging Configuration
JDBC Driver Types	Securing Web Applications	log4j Configuration
Tomcat and JDBC	HTTP Authentication	Loggers
JNDI Emulation and Pooling	Declarative security	Logger Output Hierarchy
Configuring JNDI Resources	Programmatic security	Inheriting Logging Levels
Context.xml in META-INF	Form-based Authentication	Logger Names
JDBC in Web Applications	Combined Security Mechanisms	Appenders and Layouts
Connection Pooling	SSL Characteristics	Log Analyzer Tools
Preventing Connection Leaks	SSL Handshakes	Commons Logging

Module 10 : Stack Tracing	Module 11 : Tomcat and Apache	Module 12 : Clustering
Exception Handling	Communication with Apache	Clustering Types
try, catch and finally	Advantages Using Web Server	Horizontal and Vertical Clustering
Exception information	Apache Directory Structure	Sticky Sessions
Generated Stack Trace	Configuring AJP	Load Balancing Configuration
NullPointerExceptions	Configuring mod_jk Connector	Property File Load Balancing
ClassCastExceptions	Install mod_jk	Session Sharing Backends
NumberFormat Exceptions	Create mod_jk.conf File	In-Memory Session Replication
Multiple catch clauses	Create a Worker	SimplecpCluster Configuration
User Defined Exceptions	Configure httpd.conf	Delta and BackupManager
Chained Exceptions	Proxying traffic to Tomcat	Persistent Session on File System
Reading Stack Traces	Using mod_proxy	Persistent Session in Database

# ADM300: Course JBoss Administration

Code: ADM300

#### Duration: 3 days

### Audience JBoss Administration Course

System Administrators and Web Developers who need to administer and control the JBoss Server and who need to deploy applications onto JBoss versions AS 7.\*, WildFly or EAP 6.\*.

#### **Prerequisites Course JBoss Administration**

Participants should be familiar basic computing skills like working with a command prompt and accessing the file system. Knowledge of Web Applications and other Web Servers is beneficial.

### **Realization Training JBoss Administration**

The theory is covered using presentation slides. The concepts are further explained using demo's. The theory is alternated with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification JBoss Administration**

Participants receive an official certificate JBoss Administration after successful completion of the course.

#### **Course JBoss Administration**

Advisition Advisi

# **JBoss Administration**



In this course you will learn the details of setting up, configuring, and fine-tuning the JBoss application server. Versions JBoss EAP 6.\* and AS 7.\* are used in the course. Participants in this training will learn the details of the architecture, steps involved in setting up the application server, basic deployments, production installation and deployment issues, tuning and monitoring, securing the services and clustering. This course is based on the latest JBoss Application Server though it also covers differences from the previous versions of the server. JBoss Application Server version 7.0 features many improvements over previous releases, such as centralized configuration management, easier administration, modular core and class loading, parallel deployment, improved startup times, and many new features. The course is targeted toward application support individuals, such as system administratiors, configuration management and quality assurance personnel who wish to become proficient in configuring and administrating JBoss Application Servers. But also developers of applications will benefit from the course. They also learn to understand the needs of the deployed applications as well as different deployment scenarios like clustered versus farmed.

Module 1 : Java EE Standard	Module 2 : JBoss Intro	Module 3 : Class Loading
Java EE Standard Java EE Servers Servlets and JSP's EJB Components Persistent Entities Layered Architecture Container Services Session Beans Pooling Stateless Session Beans Stateful Session Bean Life Cycle Passivation and Activation Process Java EE API's JAX-WS Java EE Web Services JAX-WS Operation JAXB for Java to XML Binding REST Web Services Java Messaging Service Message Driven Beans	JBoss AS TimeLine Kernel Taxonomy Relation to Java EE spec JBoss AS versus EAP JBoss EAP 6 Features of EAP 6 Standalone versus Domain Extensions Profiles and Subsystems Datasources Subsystem Paths Interfaces Standalone and Domain Mode Start Standalone Server Management Resources Attributes Children Descriptions	Modules Class Loading Preference Modular Class Loading Why use Modules? Static Modules module.xml Dynamic Modules Module Dependencies Dependency Example Class Loading in Deployments Dynamic Module Naming Class Loading and Subdeployments Implicit Module Dependencies Included Modules
Module 4 : JBoss Logging	Module 5 : User Management	Module 6 : JBoss Security
Logging Logging in Java log4J Basic Concepts Logging API Simple Logging JBoss Logging Supported Logging Frameworks Bootup Logging View Bootup Errors read-boot-errors Command Garbage Collection Logging Log Levels Logging Configuration Handlers and Loggers	About User Management Creating a User Using Web Console Using Command Line Interface Add user Command Arguments Specifying Realms Passing Arguments Alternate Property Files User in Single Group User in Multiple Groups Remote Access Administrator Privileges	Securing Applications Filtering Clients by Source Plain-Text Login Module Database Login Module FORM-based Login Configuring JBoss AS for SSL Creating SSL Certificates Configure SSL Connector Requiring SSL in Apps Securing JMS destinations Securing AS AS System User File System Security Java Security Manager Running Behind a Firewall

Java Database Connectivity JDBC Architecture and Operation JDBC Drivers JDBC Executing a Statement Types of DataSources Deployment of \*ds.xml Files Non XA Datasources Datasource Connection URL's Non-XA Datasources Datasource Connection URL's **Common Datasources Parameters Common Pool Parameters** Core Datasource Statistics JDBC Statistics Example MySQL Datasource Module.xml Files Module Add Command

# Module 10 : Performance Tuning

Performance Considerations Important Performance Aspects Java Performance Timeline Code Compilation and JIT Establishing Benchmarks JVM Tuning Heap Tuning and Stack Tuning Generational GC Java Object Behavior Heap Space Organization Heap Area Characteristics Measuring Garbage Collection Concurrent Mark and Sweep Tuning Garbage Collection Application Server Trimming Thread Pool Tuning Database Connection Pooling Web Services Overview Service Oriented Architecture Web Services With JAX-WS Web Services on JBoss Configure Web Services Options Modify WSDL Address WSDL Host Setting WSDL Port WSDL Secure Port Web Services Subsystem Adding Handlers Configuring Handler Chain Reload Server

What is Clustering? Clustering Terminology Clustering Types Vertical and Horizontal Sticky Sessions State Replication Clustering in Jboss HTTP Session Clustering Improved Scaling Session Beans Java Persistence API persistence.xml Hornet HA Shared Store Message Replication HTTP Connectors

# ADM400: Course IIS Administration

Code: ADM400

Duration: 3 days

# Audience IIS Administration Course

Administrators, Programmers, engineers and QA personnel who need to administer and control the IIS Server and who need to deploy applications onto IIS.

### **Prerequisites Course IIS Administration**

Participants should be familiar basic computing skills like browsing the Web and accessing the directory structure. Knowledge of Web Applications and other Web Servers is beneficial.

### **Realization Training IIS Administration**

The subjects are discussed on the basis of presentation slides and demos. The theory is interspersed with exercises. The code is tested in different browsers. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification IIS Administration**

Participants receive an official certificate IIS Administration after successful completion of the course.

#### **Course IIS Administration**

This two-day course provides students with the knowledge and skills to implement IIS

Administration. The course focuses on the IIS 7.0 architectural features that improve reliability, security, manageability and performance. The course also provides knowledge and skills to successfully install, configure, implement security, monitor, and manage Internet Information Services (IIS) 7.0 servers and Web sites. After completing this course, students will be able to describe the main features of IIS 7.0, the IIS 7.0 security model, the request processing and metabase architectures.

Module 1 : IIS Intro	Module 2 : IIS Configuration	Module 3 : Manage IIS
IIS Setup Introduction History of IIS Installing IIS Web Platform Installer Creating Websites Adding SSL Certificates SSL Wildcard Certificates	Configuration Intro Web.config Feature Delegation Application Pools Basic Application Pools Advanced Application Pools Strategy	Management Intro Troubleshooting PowerShell Web Deployment
Module 4 : IIS Extensions	Module 5 : IIS 8 New Features	
IIS Extensions Introduction Extensions Overview URL Rewrite IIS Express	IIS 8 Setup Application Initialization SNI SSL Management CPU Throttling Web Sockets	



#### Web Server Web Server Web Server Web contract Web co

Administration

# ADM500: Course Nginx Administration

Code: ADM500

Duration: 2 days

# Audience Course Nginx Administration

The course Nginx Administration is intended for system administrators and web developers who need to manage and configure the Nginx server and install applications on it.

### **Prerequisites Courses Nginx**

Participants must have basic computer skills, such as surfing the web and approaching the file system. Knowledge of web applications and other web servers is beneficial for a good understanding.

# **Realization Training Nginx**

The theory is treated on the basis of presentations. The concepts are explained with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Official Certificate Nginx Administration**

Participants receive an official certificate Nginx Administration after successful completion of the course.

#### **Course Nginx**

In the course Nginx Administration the management of the Nginx web server is discussed.





# **Nginx Administration**



Module 1 : Nginx Intro	Module 2 : Nginx Configuration	Module 3 : Nginx as Reverse Proxy
What is Nginx?	Configuration directives	Proxy connections
Nginx Architecture	Organization and inclusions	Nginx as reverse proxy
Nginx Benefits	Customize Logging	Nginx and Apache
Nginx installation	The HTTP Server	Supported upstream protocols
Install Nginx from binary	Routing and Maps	Configuring Apache and Nginx
Install Nginx from source	Mail server proxy	Nginx proxy module
Enabling modules	Setting up virtual hosts	Note on variables
Base modules	Configuring locations	Issuing redirects
Serve static content	Location lookup order	Proxying Websocket Connections
Nginx as system service	Setting up error pages	Separating content
Upgrading Nginx	Use Variables in Rewrites	X-Accel-* headers
Module 4 : Monitoring and Logging	Module 5 : Load Balancing	Module 6 : Caching
Access log Files	HTTP Load Balancing	NGINX HA Mode
Error log Files	TCP Load Balancing	The Vary: header
Log Levels	Load-Balancing Methods	Keep-alives
Specifying custom log format	Connection Limiting	Scalable Content Caching
Tracking slow requests	Clustering Types	Caching Zones
Optimizing logging	Horizonal Clustering	Caching Hash Keys
Log rotation	Vertical Clustering	Cache Bypass and Cache Performance
Log analysis by external programs	Session Persistence	Purging
Monitoring Nginx	Sticky Cookie	Sophisticated Media Streaming
Traffic Monitoring	Sticky Learn	Bandwidth Limits
Nginx stub status page	Sticky Routing	Seamless Reload
Alerts from Nginx	Connection Draining	TCP and HTTP Health Checks

#### Module 7 : Security

Controlling Access Access on IP Address Allowing CORS Limiting Connections Limiting Rate and Bandwidth Client-Side Encryption HTTP Basic Authentication Secure Links and Locations API Authentication Using JWT Creating JSON Web Keys Single Sign-On (SSO) ModSecurity Firewall HTTPS Redirects

# **DAT100: Course SQL Fundamentals**

Code: DAT100

#### Duration: 2 days

## Audience Course SQL Fundamentals

The course <u>SQL Fundamentals</u> is aimed at a wide audience of system administrators, end users of Office applications and novice programmers who want to access relational <u>databases</u> efficiently by using queries in Structured Query Language (SQL).

### **Prerequisites Course SQL Fundamentals**

To be able to participate in this course some familiarity with database systems is beneficial for the understanding.

### **Realization Training SQL Fundamentals**

The theory is treated on the basis of presentations. Demos are used to clarify the theory. There is ample opportunity to practice. The course times are from 9.30 to 16.30.

### **Certification course SQL Fundamentals**

Participants receive an official certificate SQL Fundamentals after successful completion of the course.

#### **Course SQL Fundamentals**



In the course <u>SQL Fundamentals</u> participants learn the generic ANSI SQL standard that is applicable in all database systems. SQL or Structured Query Language is an ANSI/ISO standard language for relational database management systems (DBMS). SQL is used for querying and modifying data in relational databases such as Oracle, MySQL, Microsoft Access, Microsoft SQL Server, DB2, Informix, PostgreSQL and more. Almost every DBMS has added its own extra functions to standard SQL. In the course SQL Fundamentals participants learn to extract data from a database, to analyze it and to make reports with this data. The participants learn the structure of relational databases with tables and their relationships. Attention is paid to practical skills to write SELECT queries and UPDATE Queries. You will also learn about foreign key relationships between tables and how JOIN queries can be used to retrieve data from related tables.

Module 1 : SQL Intro	Module 2 : Select Queries	Module 3 : SQL Data Definition
What is SQL? History of SQL SQL Standard SQL Parts Environment Relational Databases Normalisation Data Types Database Creation DDL Create Table Data Types Language Elements DML Insert Into SQL Errors Select Query	Query Structure SELECT FROM Options SELECT FROM SELECT DISTINCT WHERE Clause Comparison Operators Logical AND and OR Aggregate Functions LIKE Condition BETWEEN AND Condition IN Condition IS NULL Condition ORDER BY Clause GROUP BY Clause HAVING Clause	CREATE Statements Schema and Table Creation Data Types Numeric Types Binary Data Types String Data Types Temporal Data Types Integrity Constraints Keys Not Null Foreign Keys Update and Delete Check Contraint ALTER Table DROP Table
Module 4 : Functions	Module 5 : Data Manipulation	Module 6 : Joins
Standard Functions Mathematical Functions String Functions Conversion Functions Single Row Functions Case Manipulation Functions Character Manipulation Characters Numeric Functions Date Functions Converting Text Dates General Functions Multiple Row Functions Aggregate Functions Formatting Numbers	Insert Statement Data Insertion Update Statement Updating Table Rows Deleting Table Rows Deleting and Foreign Keys Transactions Commit and Rollback Implicit rollbacks Implicit rollbacks Explicit commits Savepoints Subqueries	What are Joins? ANSI Join Syntax Cross Join Inner Join Table Aliases Natural Join Left Join Right Join Full Outer Join Left Excluding Join Right Excluding Join Outer Excluding Join

# **DAT130: Course Oracle SQL Fundamentals**

Code: DAT130

#### Duration: 3 days

### Audience Oracle SQL Fundamentals Course

This course is designed for developers, database administrators and other interested parties who wish to learn and use Oracle SQL.

#### **Prerequisites Course Oracle SQL Fundamentals**

This course has no specific requirements. General knowledge of system development and databases is beneficial to a good understanding.

#### **Realization Training Oracle SQL Fundamentals**

The theory is treated on the basis of presentation slides. Demos are used to explain the theory. There is ample opportunity to practice. The course works with Oracle 11g but is also suitable for other Oracle versions. The course times are from 9.30 to 16.30.

### **Certification Oracle SQL Fundamentals**

Participants receive an official certificate Oracle SQL Fundamentals after successful completion of the course.

#### **Course Oracle SQL Fundamentals**

In the Oracle SQL Fundamentals course participants learn the syntax and the use of the SQL query language in the context of an Oracle database. After an introduction to

relational databases, the Oracle installation, environment and tooling, the basic aspects and backgrounds of SQL are discussed. In the first place, is dealt with is how SQL statements can be made and executed in Oracle. Then the various components of SQL are treated such as data definition language with CREATE TABLE, Data Manipulation Language with INSERT and UPDATE and SELECT queries. The grouping of data through GROUP and HAVING clauses is on the course program and the various functions in SQL are discussed. Also combining of data from different tables by means of joins is on the course program. Various types of joins are discussed. Finally the focus is on the application of SET operators such as UNION and INTERSECT and also the command line interface of SQL \* Plus is addressed.

Module 1 : Oracle Introduction	Module 2 : SQL Introduction	Module 3 : Data Definition	
Databases DBMS Systems Types of Database Models Entities and relationships Relational databases Oracle tools Installing Oracle Oracle versions	SQL Foundations SQL Developer Connection Navigator Creating and Using Connections SQL Worksheet Statements and execution SQL History Storing statements	CREATE TABLE DESCRIBE SQL Steps Specifying Columns ALTER TABLE NULL and Default Values DROP COLUMN DROP TABLE PURGE FLASHBACK TABLE Virtual columns	
Module 4 : Data Manipulatie	Module 5 : SQL Queries	Module 6 : Grouping	
Data Manipulation Language Transaction control Inserting rows INSERT statement Updating rows UPDATE statement DELETE statement TRUNCATE TABLE DML on virtual columns	Selecting rows SELECT statement FROM clause Specifying conditions WHERE clause Sorting with ORDER BY NULLs, FIRST, LAST Removing Doubles	Grouping Functions GROUP BY clause Nesting Filtering groups HAVING clause Operators String and Date Operators Concatenate operator Quote operator q Operators for patterns Operators for intervals BETWEEN, IN, ANY, ALL	
Module 7 : Functions	Module 8 : Joins	Module 9 : SET Operators	
Function of column values Using functions SELECT clause functions WHERE clause functions GROUP BY clause functions HAVING clause functions ORDER BY clause functions String and Arithmetic functions Nesting functions Date and Conversion functions National Language Support NLS parameters Cast	Joining Multiple Tables Normal Joins Outer Join More tables ANSI joins Inner join Natural join ANSI outer join Cross join Multiple join conditions Subselections EXISTS	SET Operatoren UNION UNION ALL INTERSECT MINUS Nesting of SET operators	
Module 10 : SQL*Plus			



# **Oracle SQL Fundamentals**

S O I

SQL\*Plus Working with SQL\*Plus Connect statement and /nolog SQL buffer Storing statements Spool files Command files Transaction control

# DAT150: Course Oracle PL-SQL Programming

Code: DAT150

Duration: 3 days

### Audience Oracle PL-SQL Programming Course

This course is designed for developers who want to use PL-SQL in applications that access an Oracle database.

#### Prerequisites Course Oracle PL-SQL Programming

In order to participate, knowledge of and experience with the SQL query language is required for this course. Experience with programming in a procedural programming language is beneficial to the understanding.

### **Realization Training Oracle PL-SQL Programming**

The theory is treated on the basis of presentation slides. Demos are used to explain the theory. There is ample opportunity to practice and theory and exercise are alternated. The course uses Oracle 11g but is also suitable for other Oracle versions. The course times are from 9.30 up and to 16.30.

### **Certification Oracle PL-SQL Programming**

Participants receive an official certificate Oracle PL-SQL Programming after successful completion of the course.

#### Course Oracle PL-SQL Programming





# **Oracle PL-SQL Programming**





In the course PL-SQL Programming participants learn to program in PL-SQL, the procedural programming language from Oracle. PL-SQL fits in with the query language SQL, but extends the capabilities of SQL and improves performance. First the position of PL-SQL amongst the different Oracle tools is discussed. Next attention is paid to the syntax of PL-SQL, the structure of PL SQL programs and statements, declarations and expressions. Also control flow in PL-SQL with if else and case statements and the various loop constructs are discussed. Next cursors are discussed and how they are used in accessing the database. Also advanced issues like the creation and calling of local and stored procedures and the use of packages are part of the subject matter. Finally, attention is paid to triggers. This course uses Oracle 11g, but where necessary, Oracle 9i, 10g and 11g features specifically indicated.

Module 1 : PL-SQL Intro	Module 2 : PL-SQL Syntax	Module 3 : Control Flow
What is PL-SQL Strenght of PL/SQL Procedural possibilities Enhanced performance Oracle Suppport tools Oracle Specifics SQL Developer	Basic Concepts PL-SQL Structure Statements Declarations Assignments %TYPE and %ROWTYPE Expressions Comments NULL Nested blocks Syntax errors	IF-THEN-ELSE statement CASE statements GOTO WHILE loops FOR loops with index FOR loops with cursor Infinite loops
Module 4 : Cursors	Module 5 : Transactions	Module 6 : Error Handling
Implicit cursor Query with SELECT INTO Query with cursor OPEN-FETCH-CLOSE FOR loops with cursor Parameters Controls Cursor attributes Self defined records	Transaction in PL-SQL Queries for changes Autonomous transactions Locking Read consistency	Exceptions RAISE EXCEPTION_INIT SQLCODE SQLERRM
Module 7 : Procedures and functions	Module 8 : Packages	Module 9 : Triggers
Local versus stored procedures Parameters DEFAULT values Removing procedures Local and stored functions Functions in SQL Operators Dependencies Recompilation Remote dependencies	Benefits of Packages Package specification Package body Calling a function in SQL RAISE_APPLICATION_ERROR DESCRIBE Removing Packages NDS Packages Standard packages for SQL	DML-triggers Statement triggers Row triggers Instead-of-triggers Trigger Limitations Auditing Data integrity Cascading triggers System Event en DDL-triggers Triggers on user logon and logoff Server error triggers Shutdown and startup triggers

# **DAT200: Course Database Design**

# Code: DAT200

#### Duration: 2 days

Price: € 1199

### Audience Course Database Design

The course Database Design in intended for Web developers, web application developers, database administrator, webmasters and web project managers.

#### Prerequisites Course Database Design

To join this course is no specific skills or knowledge is required. General knowledge of system design is helpful to a proper understanding.

#### **Realization Training Database Design**

The theory is treated using presentation slides. Demos are used to clarify the theory. There is ample opportunity to practice. The course material is in English. The course times are from 9.30 up and to 16.30.

# **Certification Database Design**

Participants receive an official certificate Database Design after successful completion of the course.

#### **Course Database Design**

In the course Database Design participants learn the techniques and considerations

for creating a well designed database. Design is a crucial activity to effectively implement a relational database. The phases of Database Design and the parts of a database are discussed. Also attention is paid how to model data and how to construct and develop of a database. Participants will learn how to discover the entities and their relations that will be mapped to tables. They will learn the principles of Entity Relationship Modeling. They will learn how to apply constraints and discover and model attribute domains. And also attention is paid to using the Unified Modeling Language UML for Database Design. Next the process of normalization, the different normal forms and the removal of redundant data is explained through practical examples. Finally a number of optimization techniques that can improve the speed of databases like the use of indexes is discussed.

Module 1 : Intro DBMS	Module 2 : Database Design	Module 3 : Entity Relationship Modeling
What is a DBMS? DBMS Abstraction Levels Data Independence Database Model Types of Databases Database Schema Conceptual Model Logical Model Physical Model SQL Language DDL and DML Language Application Interfaces Transactions DBMS Architecture	What is Database Design? Database Design Phases Benefits of Phases Conceptual Data Model Entity Relationship Model UML Model Structuring the Model Design Errors Data Errors Constraints Database Constraints Naming Schema Elements Data Interpretation CASE Tools	E-R Model Components Identification Guidelines Entities versus Entity Classes Attributes Entities versus Attributes Classification of Attributes Attribute Domains Relationships Degree of relationships Relationship Cardinalities Notation of Cardinalities Removing M:N relations Requirement Analysis Resulting ER Diagram
Module 4 : Advanced Er Modeling	Module 5 : Mapping ERD to Tables	Module 6 : UML Modeling
Weak Entity Set Generalization and Specialization Design Constraints Total and Partial Participation Disjoint Constraints Overlapping Constraints Aggregation ER Design Decisions Mapping ERD to Tables Composite Attributes Multivalued Attributes Redundancy As Tables	Entity Set Table Translation Relationship Table Translation Mapping Key Constraints Map Relationship Set to Table Combine Relationship and Entity Set Weak Entity Sets Mapping Weak Entity Sets Mapping Subclasses Table per Subclass Table per Hierarchy Discriminator Columns Joining Tables	What is UML? Structural Modeling? Core Elements Core Relationships Structural Diagrams Classes and Objects Class Diagrams Interfaces Associations Composition Generalization Dependencies
Module 7: Normalization	Module 8: Database Optimization	
What is Normalization? Unnormalized form Moving towards 1NF First Normal Form Moving to 2NF Second Normal Form Third Normal Form Other Normal Forms Benefit of Normalization Relationship Cross Tables	Optimization Process Use Ranges Denormalize Denormalization Issues Combine Tables Store Derived Data Add Indexes Index Operation Sorting Clustered Indexes	





# **Database Design**

# DAT300: Course MySQL Administration

Code: DAT300

#### Duration: 5 days

Price: € 2499

### Audience MySQL Administration Course

Persons who need to administer, monitor and support MySQL databases and servers.

### Prerequisites Course MySQL Administration

To join this course knowledge of the SQL query language and databases is needed.

### **Realization Training MySQL Administration**

The subject matter is treated on the basis of presentation slides. Demos are used to clarify the theory and exercises are used to bring the theory into practice. The course material is in English. This course fullfils the requirements for the Oracle MySQL 5 Database Administrator certification. The course times are from 9.30 up and to 16.30.

# **Certification Course MySQL Administration**

Participants receive an official certificate MySQL Administration after successful completion of the course.

#### **Course MySQL Administration**

This MySQL Database Administration course is designed for MySQL Database Administrators who have a basic understanding of a MySQL database and SQL

commands. The course provides practical experience in setting up and maintaining a MySQL server, including back up, recovery, configuration and optimization. Participants will learn to install, start and shutdown the MySQL server, learn to configure MySQL components and to use different storage engines supported in MySQL. They will also learn to maintain security of a MySQL installation via user management and access rights. Finally they will learn to work with the MySQL Administrator Graphical User Interface to perform backup and restore operations using multiple MySQL tools and to perform database replication in MySQL.

Module 1 : Intro MySQL	Module 2 : Configuring MySQL	Module 3 : MySQL Architecture
Introduction Client Program Limitations mysql MySQL Admin What is Metadata? The mysqlshow Utility The SHOW and DESCRIBE Commands The Information_Schema Database MySQL Server Options and Variables MySQL Status Variables MySQL Distributions	Installing on Windows Installing on Linux and UNIX Starting and Stopping on Windows Starting and Stopping on UNIX/Linux Log and Status Files The Default SQL Mode Time Zone Tables Some Security Issues MySQL Error Messages The SHOW Statement SQL Modes The PERROR Utility The Log The Error Log The Slow Query Log	Client/Server Overview Communication Protocols The SQL Parser and Storage Engine Tiers How MySQL Uses Disk Space and Memory Table Properties Creating and altering Tables Dropping and emptying Tables Obtaining Table Metadata Column Attributes Bit and Numeric Data Types Character String Data Types Binary String Data Types Enum and Set Data Types Temporal Data Types Auto_Increment Handling Missing or Invalid Data Values Performance Issues with Character sets
Module 4 : Storage and Locking	Module 5 : Table Maintenance	Module 6 : Backup and Recovery
Locking Concepts Explicit Table Locking Advisory Locking Preventing Locking Problems Introduction The MYISAM Engine Locking with MYISAM Tables The Merge Engine Other Engines: Archive, Memory, Federated, Blackhole, NDBCluster	Features of Innodb Transactions Referential Integrity Physical Characteristics of Innodb Tables Tablespace Configuration Log File and Buffer Configuration Innodb Status Table Maintenance Operations Check Table Repair Table Analyze Table Optimize Table MySQL Check MYISAMCHK Repairing Innodb Tables Enabling MYISAM AutoRepair	Planning and Implementing a Backup and Recovery Strategy Defining a Disaster Recovery Plan Testing a Backup and Recovery Plan The Advantages and Disadvantages of Different Methods Binary Backups of MYISAM Tables Binary Backups of Innodb Tables Recovery Import and Export Operations Exporting Using SQL Importing Using SQL Exporting from the Command Line using mysqldump Importing from the Command Line using mysqlimport
Module 7 : Security	Module 8 : Stored Procedures and Triggers	Module 9 : Optimization





# **MySQL** Administration



User Accounts Creating Users Renaming Users Changing Passwords Dropping Users Granting Privileges The User Table Connection Validation Types of Privileges Revoking Privileges Resource Limits The MySQL Database The Show Grants Command Privileges Security Issues Operating System Security Filesystem Security Filesystem Security Network Security Upgrade-related Security Issues Upgrading the Privilege Tables Security-Related SQL_Mode Values	User Variables Prepared Statements Types of Stored Routines Benefits of Stored Routines Stored Routines Features Stored Routine Privileges and Execution Security DML Triggers The Create Trigger Statement Managing Triggers	Optimization Overview Optimization Process Planning a Routine Monitoring Regime Setting Suitable Goals Identifying Candidates for Query Analysis Using Explain to Analyze Queries Meaning of Explain Output Using Explain Extended Indexes for Performance Creating and Dropping Indexes Obtaining Index Metadata Indexing and Joins Mylsam Index Caching Normalisation General Table Optimizations Myisam Specific Optimizations Innodb Specific Optimizations Other Engine Specific Optimizations Measuring Server Load System Factors Server Parameters Query Optimizer Performance The Query Cache
Module 10 : Environment Optimization	Module 11 : Scaling MYSQL	Module 12 : MySQL Workbench
Choosing the Platform Hardware Configurations Disk Issues on Linux Symbolic Links Optimizing the Operating System Exercises: Optimizing the Environment Event scheduler concepts Event scheduler configuration Creating, altering and dropping events Event scheduler monitoring Events and privileges	Partitioned tables concepts Range partitioning Hash partitioning Key partitioning List partitioning Composite partitioning or subpartitioning Maintenance of partitioned tables Using Multiple Servers Replication	Installation Connecting Server Information Service Control User Administration Privileges Health Backup and Restore Catalogs

# DAT400: Course MySQL Development

Code: DAT400

#### Duration: 5 days

## Audience MySQL Development Course

This course designed for developers assigned with application development using a MySQL database system.

#### Prerequisites Course MySQL Development

To join this course a working knowledge of SQL and databases is required.

### **Realization Training MySQL Development**

The subject matter is treated on the basis of presentation slides. Demos are used to clarify the theory and exercises are used to bring the theory into practice. The course material is in English. The course times are from 9.30 up and to 16.30.

### **Certification MySQL Development**

Participants receive an official certificate MySQL Development after successful completion of the course.

#### **Course MySQL Development**

This MySQL Developers training course is designed for MySQL Developers who have a good understanding of a MySQL database and experience of using SQL commands. The course provides further practical experience in more advanced

MySQL commands and SQL statements including Stored Procedures, Triggers and Event Scheduling. Participants will learn to use advanced features of the MySQL Client and manage the structure of databases using indexes and tables. They will also learn to write complex SQL query statements, advanced SQL expressions and SQL functions and to perform advanced Insert, Update, Delete, Replace and Truncate Operations. Also attention is paid to the importing of data into MySQL and exporting data from MySQL. The participants will learn to write and perform complex joins to access multiple tables. Finally participants will learn to create, manage and use views and work with the main storage engines to debug MySQL applications.

Module 1 : Database Concepts and MySQL	Module 2 : Installation and Configuration	Module 3 : Database Design
Features of a Relational Database Where does SQL Fit in? Database Access Why MySQL? The History of MySQL	MySQL Software MySQL Software Features Preparing to Install MySQL Available Client Software After the Download Configuring the Server Starting the Server The Initial User Accounts Verifying Server Operation Upgrading Copying a Database between Architectures Environment Variables	Developing the Design of a Database Database Entities The Primary Key Foreign Key Relationships Data Models and Normalization Second Normal Form (2NF) Third Normal Form (3NF) and beyond Translating a Data Model into a Database Design
Module 4 : mysql Command-Line Tool	Module 5 : DDL-Data Definition Language	Module 6 : Data Manipulation Language
Running the mysql Client Customizing the mysql Prompt mysql Commands Using the Help Command Some Useful mysql Options Working with a Database Examining Table Definitions Other SHOW Options	DDL and DML Overview Building Table Definitions Identifiers Column Definitions Numeric Datatypes ENUM and SET Types Date and Time Datatypes AUTO_INCREMENT UNIQUE Constraints Primary Keys Modifying Tables Foreign Keys Renaming and Dropping Tables	DDL and DML Overview Data Values: Numbers Data Values: Strings Working with NULL Values Bulk Loading of Data Bulk Data Format Working with Special Values in Bulk Data Adding New Table Rows with INSERT Copying Rows UPDATE REPLACE Removing Table Rows Transactions InnoDB: Using Transactional Processing Locking Tables
Module 7 : Queries with SELECT	Module 8 : Unions and Joins	Module 9 : Advanced SQL Techniques





# **MySQL Development**





SELECT Syntax Summary Choosing Data Sources and Destinations for SELECT Presentation of Table Data with SELECT Being Selective About Which Rows are Displayed User-Defined Variables Expressions and Functions Control Flow Operators and Functions Function Names Comparison Operators and Functions String Functions Numeric Operators and Functions Date and Time Functions Forcing Data Interpretation Miscellaneous Functions	UNION Combining Data from Two Tables Using WHERE to Choose Matching Rows INNER JOIN OUTER JOINS Multiple Tables, Fields, Joins, and Ordering SELECT * and USING Columns Advanced SQL Techniques	MySQL Pattern Matching Multipliers, Anchors, and Grouping GROUP BY Aggregates Subqueries Subquery Comparisons and Quantifiers Other Subqueries Subquery Alternatives and Restrictions InnoDB Multi-Table Updates and Deletes Building a VIEW Updatable VIEWs
Module 10 : MySQL Storage Engines	Module 11 : Utilities	Module 12 : Administering Users and DB
Storage Engine Overview Other Storage Engine Types The Basics of Commonly Used Storage Engines MyISAM Limits and Features MyISAM Data File Format InnoDB and Hardware Limitations InnoDB Shared Tablespace Configuration InnoDB Per-Table Tablespaces InnoDB Data Management MEMORY and FEDERATED MERGE and ARCHIVE	Client Overview Specifying Options for Command-Line Clients Client Option Files Checking Tables with myisamchk and mysqlchk Using myisamchk and mysqlchk for Repairs mysqlshow and mysqlimport Using mysqldump The Query Browser MySQL Query Browser: Deeper MySQL Administrator: Basic Operations MySQL Administrator: Monitoring the Server and User Administration Third Party Tools	The Server-Side Programs Starting the MySQL Server Using SET for Server Options Table Management Server Log Files mysqladmin Backup and Restore Miscellaneous Functions User Account Management Understanding User Privileges User Account Rights Management User Account Privileges Managing Access to the Database Environment
Module 13 : Database Programmability	Module 14 : Performance Tuning	Module 15 : MySQL Programming Interfaces
Stored Routines: Basic Concepts Routine Creation and Use Flow Control Statement Writing Blocks of Code Triggers Stored Routines, Triggers Binary Log Table HANDLERs Prepared Statements	Hardware Limitations Optimizing the MySQL Server's Interaction with the External World Adjusting the MySQL Server Configuration Optimizing Your Database Optimizing Queries The Use of Indexes to Support Queries Thinking about JOIN Queries Query Sorts, Indexes, and Short-Circuiting INSERT, UPDATE, DELETE, and Talkie Locks Some General Optimizations Optimizations Specific to MyISAM Optimizations Specific to InnoDB	Database Application Architectures Connecting MySQL to ODBC Connecting MySQL to MS/Office and MS/Access Connecting to MySQL from Perl Programming Perl to MySQL Connecting to MySQL from PHP Programming PHP to MySQL

# DAT440: Course MongoDB Data Access

Code: DAT440

Duration: 2 days

# Audience MongoDB Data Access

The course MongoDB Data Access is intended for developers and database administrators who want to learn how the NoSQL database MongoDB can be accessed and managed.

#### Prerequisites MongoDB Data Access

Participants must have basic computer skills. Knowledge of data access with other databases is beneficial for good understanding.

### **Realization Training MongoDB Data Access**

The theory is treated on the basis of presentations. The concepts are explained with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Official Certificate MongoDB Data Access**

Participants receive an official certificate MongoDB Data Access after successful completion of the course.

# Course MongoDB Data Access

The course MongoDB Data Access discusses access and administration of the No SQL Database MongDB.



35.1 GB	4 Databases	<pre>HongoOB Server Information {     "process": "mongod",     "pid": 3309,     "connections": {</pre>
Database Name	Size .	"curnent": 2, "available": 817
e la	33.9 GB	), "locks": (
wat .	976.0 MB	"admin": { "timeAcquiringMicros": {
admin	208.0 MB	"r": 3014, "w": 0
ocal	1 byte	<b>b</b>

# MongoDB Data Access



Module 1 : Intro MongoDB	Module 2 : MongoDB Data Model	Module 3 : MongoDB Queries
MongoDB Design Philosophy MongoDB Architecture Document Oriented Databases Speed, Scalability, and Agility Non-Relational Approach No SQL Advantages and Disadvantages JSON-Based Document Store Performance vs. Features Running the Database Anywhere MongoDB Installation Starting MongoDB JavaScript Console	JSON and BSON The Identifier _id Capped Collections Polymorphic Schema's Object Orientation Schema Evolution MongoDB Drivers MongoDB Shell Using Authentication Using Authentication Using Authorization Controlling Network Access MongDB Cloud Manager	Create Database in MongoDB Create Collection Insert into Collection Insert Multiple Documents Id Field Find One, All and Some Query Filter Regex Indices MapReduce Projections Pagination
Module 4 : MongoDB Administration	Module 5 : Connecting with Languages	Module 6 : Advanced Operations
Renaming collections Viewing Collections Stats Viewing Database stats mongostat and mongotop utilities Killing processes Setting up users Replica Set Configurations Clustering and Load Balancing Fault tolerance and Backups	Python connecting with PyMongo Aggregation with PyMongo MapReduce with PyMongo Java Clients Update and Delete with Java Query and insert with Java MongoDB and JPA MongDB and REST MongoDB and NodeJS	Atomic find Atomic modify Atomic counters Server side scripts Capped collection cursors Converting collections Storing binary data Storing large data Storing data to GridFS

# DAT500: Course PostgreSQL Administration

Code: DAT500

Duration: 4 days

### Audience PostgreSQL Administration Course

The course PostgreSQL Administration is intended for persons who need to administer, monitor and support PostgreSQL databases and servers.

#### Prerequisites Course PostgreSQL Administration

To join this course knowledge of the SQL query language and databases is needed.

### Realization Training PostgreSQL Administration

The subject matter is treated on the basis of presentations. Demos are used to clarify the theory and exercises are used to bring the theory into practice. The course times are from 9.30 up and to 16.30.

# **Certification Course PostgreSQL Administration**

Participants receive an official certificate PostgreSQL Administration after successful completion of the course.

# Course PostgreSQL Administration

The PostgreSQL Database Administration course is designed for PostgreSQL Database Administrators who have a basic understanding of the PostgreSQL database and SQL commands.





# PostgreSQL Administration





Module 1 : PostgreSQL Intro	Module 2 : PostgreSQL Architecture	Module 3 : Server and Database Objects
What is PostgreSQL? PostgreSQL Features User-defined types Table inheritance Sophisticated locking mechanism Foreign key referential integrity Nested Transactions and Savepoints Multi-version Concurrency Control Asynchronous replication Install and Connect to PostgreSQL pgAdmin Application and psql Shell	Shared Memory Shared and WAL Buffer PostgreSQL Process Types Postmaster Daemon Process Background Processes Backend Processes Client Processes Database Structure Create User Database What are Tablespaces? What is Vacuum?	Server Service Database Object Table Object and Schema PostgreSQL Tablespaces pg_default and pg_global View as Virtual Tables Functions and Operators Server Configuration Logging Parameters Memory Parameters WAL Parameters
Module 4 : Backup and Restore	Module 5 : Indexes	Module 6 : Database Management
PostgreSQL Backup Tools pg_dump and pg_dumpall Point-in-Time Database Restoration Setting up WAL archiving Data and/or Structure pg_dump Parameters Backup Object Definitions pg_restore and psql Ignore and Stop on Errors pg_store Parameters Restore only Structure	What are Indexes? Index Types B-Tree, Hash, GiSY GIN and BRIN Index Differences Create and Drop Index List indexes Unique Index Index on Expression Partial index and Reindex Multicolumn Indexes	Options to Create Databases Modify Databases Rename Databases Change Owner and Tablespace Change Session Defaults Delete Databases Check Activity with pg_stat_activity Copy a Database Using pg_dump Get Database Object Sizes Using pg_size_pretty
Module 7 : Role Management	Module 8 : Securing PostgreSQL	Module 9 : Cluster Management
Access Permissions pg_hba.conf file Creating PostgreSQL Roles Privileges and Restrictions superuser Login Privilege pg_roles System Catalog Role Attributes Groups Role Membership User Role and Group Inheritance	Client Authentication Control Rule Specification Server Configuration Changing Parameters Role Strategies Super User Management Data Encryption One and Two Way Encryption pg_crypto Logging pg_stat_statements Module	High Availability Cluster Performing Replication Primary Server Promoting Standby Server Load Balancing HAProxy Configuration Xinetd Setup HAProxy in ClusterControl Reslaving Standby Server Chained Replication Keepalived

# DAT600: Course PostgreSQL Development

Code: DAT600

# Duration: 4 days

Module 1 : PostgreSQL Intro	Module 2 : Server and Database Objects	Module 3 : PostgreSQL Queries
What is PostgreSQL?	Server Service	Basic Select Queries
PostgreSQL Features	Database Object	GroupBy and Having
User-defined types	Table Object	Fetch, In, Between
Table inheritance	Schema	Union and Intersect
Sophisticated locking mechanism	PostgreSQL Tablespaces	Cross and Natural Join
Foreign key referential integrity	pg_default and pg_global	Grouping Sets
Nested Transactions and Savepoints	View as Virtual Tables	Any and All
Multi-version Concurrency Control	Updatable Views	Cubes and Rollups
Asynchronous replication	Functions and Operators	Subqueries
Tablespaces	Casts	Except
Install and Connect to PostgreSQL	Sequences	IsNull
pgAdmin Application and psql Shell	Extensions	Upsert
Module 4 : PostgreSQL Functions	Module 5 : Indexes	Module 6 : Views
Aggregate Functions	What are Indexes?	Views as Virtual Tables
Sum, Count, Max and Min	Create Index	Managing Views
Date and Time Functions	Drop Index	Creating Views
String Functions	List indexes	Changing Views
RegExp Functions	Index Types	Removing Views
Math Functions	Unique Index	Creating Updatable Views
Power, Ceiling, Round	Index on Expression	Materialized Views
Window Functions	Partial index	Refreshing Data
Row_Number and Rank	Reindex	With Check Option Views
First and Last_Value	Multicolumn Indexes	Recursive Views
Module 7 : Triggers	Module 8 : Stored Procedures	Module 9 : Language Interfaces
Trigger Intro	Stored Procedures Intro	PHP PostgreSQL Access
Triggers and Events	PL/pgSQL Block Structure	PDO Driver
SQL Standard Triggers	Errors and Messages	Transactions from PHP
Postgre Triggers	Create Function	Working with Blobs
Creating Triggers	Function Parameters	Python PostgreSQL Access
Create Trigger Statement	Function Overloading	psycopg Database Adapter
Managing Triggers	Function Returning Table	Stored Procedures From Python
Modifying Triggers	Variables and Constants	Java PostgreSQL Access
Disabling Triggers	IF and CASE and Loop Statements	PostgreSQL JDBC Driver
Removing Triggers	Cursors	Calling Stored Functions

# **DAT700: Course APEX Fundamentals**

Code: DAT700

#### Duration: 3 days

### Audience APEX Fundamentals Course

This course is intended for developers who want to develop APEX Web Applications based on an Oracle database.

#### **Prerequisites Course APEX Fundamentals**

To participate in this course basic knowledge of HTML and CSS is desirable and experience with SQL and PL-SQL programming in an Oracle environment is required.

#### **Realization Training APEX Fundamentals**

The course is given with Oracle 11g and APEX 4.2. The theory is treated on the basis of presentation slides and the concepts are explained with demos. The theory is interspersed with exercises. The course times are from 9.30 up and to 16.30.

## **Certification APEX Fundamentals**

Participants receive an official certificate APEX Fundamentals after successful completion of the course.

### **Course APEX Fundamentals**

In this course, participants learn to develop Web applications with the tool Oracle

Application Express, APEX, which is included with the Oracle 11g database software.

First attention is paid to the structure of the APEX development environment and the structure of APEX Web applications. The course is for a large part concerned with the creation of screens and reports. You will learn how to quickly develop a Web Application for these purposes by using standard components and wizards. The use of standard components is discussed in detail. Also the navigation between pages is addressed and it is discussed how to customize the web application layout. Finally, attention is paid how you can achieve specific functionality and modifications of the standard components by using PL-SQL.

Module 1 : APEX Intro	Module 2 : Applications	Module 3 : Reports and Forms
APEX Architecture APEX installation APX configuration Application Builder Basics of Oracle Web Applications Managing Workspaces User Roles Administration Starting with Oracle APEX Utilities Language issues SQL Workshop	From scratch From Spreadsheet Import Export Shared Components List of Values Types of LOV's Items and Buttons Application Deployment	Reports from scratch Interactive reports Report headers Report templates Break formatting Column formatting Tabular forms Master detail forms Editing form attributes Validations Computations
Module 4 : Pages and Regions	Module 5 : Navigation and Layout	
Pages Page Definition Page Processing Page Rendering Page Manipulations Page Zero Regions Region Definition Calendars Charts	Tabs Navigation Bars Breadcrumbs Lists Trees Themes Templates Template type and Classes Template Files Template Files Template Subscriptions	



# DAT800: APEX Advanced Course

# Code: DAT800

Duration: 3 days

# Audience Advanced Course APEX

This course is intended for developers who already have some experience in developing web applications with APEX on an Oracle database.

#### **Prerequisites APEX Advanced Course**

To participate in this course basic knowledge of HTML, CSS and JavaScript is required as well as basic knowledge and experience in developing with APEX.

### **Realization Training APEX Advanced**

The course is given with Oracle 11g and APEX 4.2. The theory is treated on the basis of presentation slides and the concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 up and to 16.30.

## **Certification APEX Advanced**

Participants receive an official certificate APEX Advanced after successful completion of the course.

#### **APEX Advanced Course**

In this APEX Advanced course developers learn advanced features of the

development with APEX. During the course, participants develop an advanced APEX application. The course covers advanced aspects of page processing. Also various aspects of security and authentication are addressed. Adjustments to the look and feel of the application and interaction capabilities of the interface with Ajax functionality are also discussed. For this advanced functionality APEX offers possibilities in combination with the use of JavaScript and HTML. Furthermore the generation of PDF from APEX applications is treated and the use of Web Services is discussed. Finally the course also looks ahead to the new features of APEX 5.0.

Module 1 : Advanced Page Processing	Module 2 : Security	Module 3 : Ajax and PDF Generation
File handling Public APEX API's Internal APEX API's Advanced Interactive Reports Parameterized Reports Multi Valued Check Boxes Performance Debugging	Implementing Security Access Control Authorization and Authentication Authentication schemes Creating authentication scheme APEX account credentials Custom authentication LDAP en SSO Sessions Session State Protection (SSP)	What is Ajax? XMLHttp Requests Partial Page updates Callback functions Dynamic HTML and JavaScript PDF from reports XSL-FO integration FOP libraries
Module 4: Web Services	Module 5 : Apex 5 features	
What are Web Services? SOAP and WSDI standard Creating Web Service References Creating a New Application Specifying Proxy Server Address Web Service Reference from WSDL Web Service Reference Manually Using Web Services	New Page Designer New Navigation methods Model pages File management Calendar enhancements New themes New templates	



# **APEX Advanced**

# DSC100: Course Data Analysis with Python

Code: DSC100

Duration: 4 days

Module 1 : Python Language Syntax	Module 2 : Functions and Modules	Module 3 : Classes and Objects
Python Features	Pass by Value and Reference	Creating Classes
Running Python	Scope of Variables	Creating and Using Objects
Anaconda Distribution	EFAP principle	Accessing Attributes
IPython Shell	What are comprehensions?	Property Syntax
Interactive and Script Mode	Lambda Operator	Constructors and Destructors
Python Data Types	Filter, Reduce and Map	Encapsulation
Numbers and Strings	List comprehensions	Inheritance
Sequences and Lists	Set and Dictionary comprehensions	super Keyword
Sets and Dictionaries	Creating and Using Modules	Checking Relationships
Python Flow Control	import Statement	issubclass and isinstance
Exception Handling	fromimport Statement	Overriding Methods
Module 4 : Numpy	Module 5 : Pandas	Module 6 : Data Manipulation
NumPy Numerical Types	Pandas DataFrame	Indexing Data Frames
Data Type objects	Import Data	Joc and Joc Accessor
dtype attributes	Inspect Data	Slicing and Indexing a Series
Slicing and Indexing	Data Visualization	Filtering with Boolean Series
Array comparisons	DataFrame Data Types	Zeros and NaNs
Manipulating array shapes	Indexing and selection	all and any Nonzeros
Stacking and Splitting arrays	Data operations in pandas	Using map Function
any(),all(), slicing, reshape()	Missing Data	Hierarchical Indexing
Manipulating array shapes	Hierarchical Indexing	Rearranging Data
Methods of ndarray	Plotting with Pandas	Reshaping by Pivoting
Views versus copies	Combining Datasets	Transformation and Aggregation
ravel(),flatten(),transpose()	Exploratory Data Analysis	Grouping Data
Module 7 : MatplotLib	Module 8 : Time Series	Module 9 : SciKitLearn Essentials
Simple Plots	Indexing Pandas Time Series	SkiKit Learn library
Plot format String	Reading and Slicing Times	Machine learning essentials
Subplots	Using a DatetimeIndex	Supervised and Unsupervised
Histograms	Reindexing the Index	Feature matrix
Logarithmic Plots	Separating and Resampling	Target array
Scatter plots	Rolling mean and Frequency	Estimator API
Fill between	Resample and Roll with it	Hyperparameters
Legend and Annotations	Manipulating Time Series	Fit method
Three Dimensional Plots	Method chaining and Filtering	Predict method
Contour Plots	Missing values and Interpolation	Model Selection
Transformations	Time Zones and Conversion	Linear Regression
Projections	Plotting Time Series	Logistic Regression

# DSC200: Course Data Analysis with R

Code: DSC200

Duration: 3 days

### Audience Course Data Analysis with R

The course Data Analysis with  $\mathbf{R}$  is intended for Big Data analysts and scientists who want to use R to analyze their data and to make static analyzes.

### Prerequisites Course Data Analysis with R

Experience with **programming** is beneficial for a good understanding but is not required.

### **Realization Training Data Analysis with R**

The theory is treated on the basis of presentations and examples. The concepts are explained with demos. The theory is interspersed with exercises and there is ample time to practice. R Studio is used as a development environment. The course times are from 9.30 to 16.30.

#### **Certification Data Analysis with R**

The participants will receive an official certificate Data Analysis with R after successful completion of the course.

In the course Data Analysis with R you will learn programming in the language R and how you can use R for effective data analysis and visualization. R has become a

standard platform for data analysis and making graphs and you can perform a huge set of statistical procedures that are not available in other statistical programs. You first learn how to install and configure R. Next you will learn how to quickly gain insight into the data by means of graphs and transformations. The input of data from different sources is treated. Also data types of R, such as vectors, arrays, matrices, lists, data frames and factors are discussed. As well as control flow in R with the family of apply functions. The course also discusses statistical analysis models such as linear and non-linear models, variable transformations and regressions. Finally, attention is paid to how to present results through graphs, reports or interactive dashboards. All this is explained with many practical examples and can also be applied to cases that are brought in by the course participants.

Module 1 : Intro R	Module 2 : Graphics and Plots	Module 3 : Transformations
Overview of R History of R Installing R The R Community R Development R Studio R Console R Style Using R Packages Cheatsheets R Syntax R Objects	ggplot2 Graphics Devices and Colors High-Level Graphics Functions Low-Level Graphics Functions Graphical Parameters Controlling the Layout Changing Plot Types Quick Plots and Basic Control Aesthetics Changing Plot Types Labels Themes and Layout	dplyr R Functions Functions for Numeric Data Scoping Rules mutate arrange group by summarize select filter joining dataframe
Module 4 : Presentation	Module 5 : Data Cleaning	Module 6 : Date Times
rmarkdown Reproducible research Reporting Sharing results Repetitive Tasks Family of apply Functions apply Function lapply Function sapply Function tapply Function	tidyr spread gather seperate unite Logical Data Missing Data Character Data Duplicate Values NA's	Time and Date Variables lubridate Setting a datetime Getting values from a datetime strftime Command as.Date function Datetimes Calculations difftime Command Time Series Analysis
Module 7 : Data Import	Module 8 : Linear Models	Module 8 : Non-Linear Models
R Datasets Data.Frames Importing CSV Files Import from Text Files Import from Excel Import from Spss or SAS Connecting to a database Connecting to a cluster Databases and ODBC dbplyr	What is a model? Statistical Models in R How to evaluate a model? How to use a model? Simple Linear Models logistic regression linear regression R squared p values confidence intervals	Decision Trees random forest boosting overfiting Optional material : Interactive dashboards with Shiny Web Scraping Writing packages Spark Functional programming



# Data Analysis with R

# DSC300: Course Data Science with Python

Code: DSC300

# Duration: 3 days

Module 1 : Python Language Syntax	Module 2 : Functions and Modules	Module 3 : Classes and Objects
Python Features	Pass by Value and Reference	Creating Classes
Running Python	Scope of Variables	Creating and Using Objects
Anaconda Distribution	EFAP principle	Accessing Attributes
IPython Shell	What are comprehensions?	Property Syntax
Interactive and Script Mode	Lambda Operator	Constructors and Destructors
Python Data Types	Filter, Reduce and Map	Encapsulation
Numbers and Strings	List comprehensions	Inheritance
Sequences and Lists	Set and Dictionary comprehensions	super Keyword
Sets and Dictionaries	Creating and Using Modules	Checking Relationships
Python Flow Control	import Statement	issubclass and isinstance
Exception Handling	fromimport Statement	Overriding Methods
Module 4 : Numpy	Module 5 : Matrices Calculations	Module 6 : MatplotLib
NumPy Numerical Types	ufuncs	Simple Plots
Data Type objects	Creating matrices	Plot format String
dtype attributes	Universal functions	Subplots
Slicing and Indexing	Arithmetic functions	Histograms
Array comparisons	Bitwise functions	Logarithmic Plots
Manipulating array shapes	Comparison functions	Scatter plots
Stacking and Splitting arrays	Fancy indexing	Fill between
any(),all(), slicing, reshape()	at() method	Legend and Annotations
Manipulating array shapes	Inverting matrices	Threedimensional Plots
Methods of ndarray	Finding eigenvalues	Contour Plots
Views versus copies	Singular value decomposition	Transformations
ravel(),flatten(),transpose()	Pseudo inverse	Animation
full() and full_like() Functions	Determinants	Projections
Module 7 : Pandas	Module 8 : Time Series	Module 9 : SciKitLearn Essentials
Pandas DataFrame	Indexing Pandas Time Series	SkiKit Learn library
Import Data	Reading and Slicing Times	Machine learning essentials
Inspect Data	Using a DatetimeIndex	Supervised and Unsupervised
Data Visualization	Reindexing the Index	Feature matrix
DataFrame Data Types	Separating and Resampling	Target array
Indexing and selection	Rolling mean and Frequency	Estimator API
Data operations in pandas	Resample and Roll with it	Hyperparameters
Missing Data	Manipulating Time Series	Fit method
Hierarchical Indexing	Method chaining and Filtering	Predict method
Plotting with Pandas	Missing values and Interpolation	Model Selection
Combining Datasets	Time Zones and Conversion	Linear Regression
Exploratory Data Analysis	Plotting Time Series	Logistic Regression
# **DSC400: Course Advanced R for Data Science**

Code: DSC400

#### Duration: 3 days

#### Audience Course Advanced R for Data Science

The course Advanced R for Data Science is intended for data analists and data scientists who want to use the R libraries for modeling and machine learning.

#### Prerequisites training Advanced R for Data Science

To participate in this course knowledge and experience with the programming language R for Data Analysis is required. Prior knowledge with regard to statistical methods and algorithms is beneficial for the understanding.

#### **Realization Advanced R for Data Science course**

The theory is treated on the basis of presentations. Illustrative demos clarify the concepts. The theory is interspersed with exercises and case studies. The course times are from 9.30 to 16.30.

#### **Official Certificate Advanced R for Data Science**

Participants receive an official Advanced R for Data Science certificate after successful completion of the course.

#### **Course Advanced R for Data Science**

In the course Advanced R for Data Science you will learn how to apply the R





## Data Science with R

# Data Science with R

language and R libraries for modeling projects and machine learning. First the fundamentals of R and a number of important libraries are discussed in a review. Subsequently the principles of machine learning and the differences between supervised and unsupervised learning are discussed. Linear regression and logistic regression and the differences between them are treated. The course also covers how models can be checked for accuracy by looking at summaries, coefficients and plots. Next attention is paid how functional programming techniques can be applied in R. Other solutions for iteration through the various map and other functions are discussed. Attention is also paid how to use Apache Spark from R by means of a distributed data frame implementation with operations like selection, filtering and aggregation. Visualization in interactive web applications directly from R via the Shiny package is also on the program. The course is concluded with the treatment of various Machine Learning algorithms.

Module 1 : R Review	Module 2 : Linear Regression	Module 3 : Logistic Regression
R Data Types	Machine Learning	Compare with Linear Regression
Data Frames	Supervised Learning	Explore with Graphics
Factors	Unsupervised Learning	Checking Model
Rmarkdown	Check Model	Using Summary
tidy package	Using Summary	Using Coefficients
Functions in R	Using Coefficients	Calculate Probabilities
Apply functions	Correlation R	Making Predictions
Statistics	R Squared	Confusion Matrix
R Data Files	F Test	Accuracy
Using dplyr Package	Check Model Graphically	Precision and Recall
Plotting with ggplot2	Check Residuals	ROC Curve
Module 4 : Functional R	Module 5 : Sparklyr Intro	Module 6 : Shiny
Solving Iteration	Spark Session	Web Applications
purr package	Copy data into Spark	Shiny Architecture
library tidyverse	File Setup	Shiny Server
map Functions	Load data	Ul and Server
Parameters of map	Spark SQL	Input Object
.x as placeholder	Store Data	Output Object
map_lgl Function	Using dplyr	Reactivity
map_int and map_char	showquery()	Render Options
map2 Function	Spark DataFrame Functions	Shiny Functions
Other iteration functions	sdf_pivot()	Shiny Layout
Combine purr with dyplr	Feature Transformers	Shiny Dashboard
walk Function	Distributed R	Shiny Performance

#### Module 7 : Machine Learning

Principal Component Analysis Least Squares Polynomial Fitting Constrained Linear Regression K-Means Clustering Support Vector Machines Conditional Random Fields Decision Trees TD Algorithms

# DSC500: Course Data Analysis with PySpark

Code: DSC500

#### Duration: 4 days

Module 1 : Python Primer	Module 2 : Spark Intro	Module 3 : HDFS
Python Syntax	What is Apache Spark?	Hadoop Environment
Python Data Types	Spark and Python	Environment Setup
List, Tuples, Dictionaries	PySpark	Hadoop Stack
Python Control Flow	Py4j Library	Hadoop Yarn
Functions and Parameters	Data Driven Documents	Hadoop Distributed File System
Modules and Packages	RDD's	HDFS Architecture
Comprehensions	Real Time Processing	Parallel Operations
Iterators and Generators	Apache Hadoop MapReduce	Working with Partitions
Python Classes	Cluster Manager	RDD Partitions
Anaconda Environment	Batch versus Stream Processing	HDFS Data Locality
Jupyter Notebooks	PySpark Shell	DAG (Direct Acyclic Graph)
Module 4 : SparkConf	Module 5 : SparkContext	Module 6 : RDD's
SparkConf Object	Main Entry Point	Resilient Distributed Datasets
Setting Configuration Properties	Executor	Key-Value pair RDDs
Uploading Files	Worker Nodes	Parallel Processing
SparkContext.addFile	LocalFS	Immutability and Fault Tolerance
Logging Configuration	SparkContext Parameters	Transformation Operations
Storage Levels	Master	Filter, groupBy and Map
Serialize RDD	RDD serializer	Action Operations
Replicate RDD partitions	batchSize	Caching and persistence
DISK_ONLY	Gateway	PySpark RDD Class
MEMORY_AND_DISK	JavaSparkContext instance	count, collect, foreach,filter
MEMORY_ONLY	Profiler	map, reduce, join, cache
Module 7 : Spark Processing	Module 8 : Broadcast and Accumulator	Module 9 : Algorithms
SQL support in Spark	Performance Tuning	Iterative Algorithms
Spark 2.0 Dataframes	Serialization	Graph Analysis
Defining tables	Network Traffic	Machine Learning API
Importing datasets	Disk Persistence	mllib.classification
Querying data frames using SQL	MarshalSerializer	Random Forest
Storage formats	Data Type Support	Naive Bayes
JSON / Parquet	Python's Pickle Serializer	Decision Tree
GraphX	DStreams	mllib.clustering
GraphX library overview	Sliding Window Operations	mllib.linalg
GraphX APIs	Multi Batch and State Operations	mllib.regression

# **DSC700: Course Hadoop for Big Data**

Code: DSC700

#### Duration: 3 days

#### Audience Course Hadoop for Big Data

The course <u>Hadoop for Big Data</u> is intended for developers, data analysts and others who want to learn how to process data with Hadoop.

#### Prerequisites training Hadoop for Big Data

To participate in this course prior knowledge of programming in Java and databases is beneficial for the understanding. Prior knowledge of Java or Hadoop is not necessary.

#### **Realization Course Hadoop for Big Data**

The theory is treated on the basis of presentations. Illustrative demos are used to clarify the covered concepts. There is ample opportunity to practice and theory and practice are interchanged. The course times are from 9.30 to 16.30.

#### **Official Certificate Course Hadoop for Big Data**

Participants receive an official certificate Hadoop for Big Data after successful completion of the course.

#### **Course Hadoop for Big Data**

In the course Hadoop for Big Data participants learn how to use Apache Hadoop for

the storage and processing of large amounts of data. Hadoop uses a simple programming model in a distributed environment over a cluster of computers. The architecture of Hadoop is explained in depth. The Hadoop Distributed File System (HDFS) is used as file system within a Hadoop cluster. HDFS is a horizontal scalable file system that is stored on a cluster of servers. The data is stored in a distributed manner and the file system automatically ensures replication of data over the cluster. An important algorithm for the processing of data is the MapReduce algorithm and this is given extensive attention. Finally attention is paid to tools and utilities that are often used in combination with Hadoop such as Zookeeper, Scoop, Ozie and Pig.

Big

Module 1 : Hadoop Intro	Module 2 : Java API	Module 3 : HDFS
Big Data Handling No SQL Comparison to Relational Databases Hadoop Eco-System Hadoop Distributions Pseudo-Distributed Installation Namenode Safemode Namenode High Availability Secondary Namenode Hadoop Filesystem Shell	Create via Put method Read via Get method Update via Put method Delete via Delete method Create Table Drop Table Scan API Scan Caching Scan Batching Filters	Hadoop Environment Hadoop Stack Hadoop Yarn Hadoop Distributed File System HDFS Architecture Parallel Operations Working with Partitions RDD Partitions HDFS Data Locality DAG (Direct Acyclic Graph)
Module 4 : Hbase Key Design	Module 5 : MapReduce	Module 6 : Submitting Jobs
Storage Model Querying Granularity Table Design Tall-Narrow Tables Flat-Wide Tables Column Family Column Qualifier Storage Unit Querying Data by Timestamp Querying Data by Row-ID Types of Keys and Values SQL Access	MapReduce Model MapReduce Theory YARN and MapReduce 2.0 Daemons MapReduce on YARN single node MapReduce framework Tool, ToolRunner and GenericOptionsParser Running MapReduce Locally Running MapReduce on Cluster Packaging MapReduce Jobs MapReduce CLASSPATH Decomposing into MapReduce Workflow	MapReduce Job Using JobControl class Joining data-sets User Defined Functions (UDF) Logs and Web UI Input and Output Formats Anatomy of Mappers Reducers and Combiners Partitioners and Counters Speculative Execution Distributed Cache YARN Components
Module 7 : Hadoop Streaming	Module 8 : Utilities	Module 9 : Hive
Implement a Streaming Job Contrast with Java Code Create counts in Streaming App Text Processing Use Case Key Value Pairs \$yarn command Using Pipes	ZooKeeper Scoop Introduce Oozie Deploy and Run Oozie Workflow Pig Overview Execution Modes Developing Pig Script	Hive Overview and Concepts Hive Clients Table Creation and Deletion Loading Data into Hive Partitioning Bucketing Joins



HDES



# Hadoop for Big Data

# **INT100: Course Internet and Intranet Concepts**

Code: INT100

Duration: 2 days

#### **Audience Internet and Intranet Concepts Course**

This course is designed for those who wish to learn about the background and operation of the Internet and Intranet.

#### **Prerequisites Course Internet and Intranet Concepts**

To join this course is no specific skills or knowledge is required.

#### **Realization Training Internet and Intranet Concepts**

The concepts are treated with the help presentation slides. A demo Web site is used to clarify the concepts. Attention is also paid to hands-on exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Internet and Intranet Concepts**

Participants receive an official certificate Internet and Intranet Concepts after successful completion of the course.

#### **Course Internet and Intranet Concepts**



Fire

Firewall



## **Internet and Intranet** Concepts



The course discusses the concepts and protocols of the Internet and Intranets.

Attention is paid to the role of markup languages like XML and HTML. Further subjects are CSS and Dynamic HTML, client-side JavaScript scripting, the various possibilities of server-side processing such as ASP.NET, PHP or Java Servlets and the inclusion of reusable components such as applets in Web pages and ASP.NET controls. Finally, attention is paid to recent developments relating to Rich Web User Interfaces and Web Services.

Module 1 : Internet Protocols	Module 2 : HTTP Web Servers	Module 3 : HTML Markup Language
Basic Web Concepts TCP/IP Protocols and OSI Model RFC's IP Overview IP Header IP Address Scheme Domain Name System (DNS) File Transfer Protocol (FTP) SMTP SMTP, DNS and POP Topology SMTP Functions DHCP Protocol DHCP Operation	HTTP Protocol Web Servers Web Browsers Uniform Resource Locators URL's HTTP Request Messages HTTP Request Example HTTP Response Messages HTTP Response Example HTTP Request Headers HTTP Response Headers HTTP Status Codes MIME Types	Structuring business requirements HTML Markup Language HTML Pages HTML Page Structure Basic HTML Elements HTML Attributes HTML Attributes HTML Tables HTML Tables HTML Images HTML Frames HTML Forms HTML Form Structure Input Tags
Module 4 : XML Markup Language	Module 5 : Internet Security	Module 6 : CSS
XML Markup Language XML versus HTML Roots of XML Markup Languages Benefits of XML XML Elements XML Attributes Well Formed Documents Valid Documents XHTML Markup Language HTML versus XHTML XHTML Declarations Advantages of XHTML	Parts of Internet Security Security Threats Security Issues HTTP Basic Authentication HTTP Digest Authentication Encryption Types Symmetric Encryption Asymmetric Encryption Hash Encryption Signing Secure Sockets Layer SSL Handshakes Secure HTTP (HTTPS) Digital Certificates Public Key Cryptography Issuing and Using Certificates DeMilitarized Zone	What is CSS? Defining CSS Applying CSS CSS Positioning Absolute Positioning Relative Positioning Attributes CSS Visibility Property CSS Z-Index Property CSS and XML
Module 7 : Client Side JavaScript	Module 8 : Dynamic HTML	Module 9 : Server Side Scripting
Dynamic Content Techniques Client Side Scripting JavaScript's History JavaScript Characteristics Variables JavaScript Types Numbers Strings Booleans Arrays Functions Objects Event Handler Validation in JavaScript	What is DHTML? DHTML Technologies HTML DOM HTML DOM Example Browser Object Model Element Access Important Elements Important Properties Event Handlers	Servlets Reading Form Data in Servlets Java Server Pages Simple JSP Page PHP Scripts Using PHP PHP \$_POST Global Variable PHP \$_GET Global Variable Why Session Tracking? Session ID's Session Tracking Mechanisms

Module 10 : Web Services	Module 11 : Rich Internet Applications
What is a Web Service? XML Transport with Web Services RPC versus Document Style What is SOAP? Structure SOAP Message Skeleton SOAP Message SOAP Messages as Payload SOAP Header What is WSDL? Basic Structure WSDL? WSDL and Code Generation Service Orientation	Traditional Webapps Problems Rich Internet Applications What is Flex? Flex Framework Flex Architecture How Flex works ActionScript Flash Player MXML Example SilverLight XAML Drawing with XAML HTML5

# **INT200: Course HTML Fundamentals**

Code: INT200

#### Duration: 1 day

Price: € 450

#### Audience HTML Fundamentals Course

The course HTML Fundamentals is intended for anyone who wants to learn how to create HTML web pages that are part of web applications on the Internet.

#### **Prerequisites Course HTML Fundamentals**

Knowledge of the basic aspects of the Internet and the use of a browser is required to participate in this course.

#### **Realization Training HTML Fundamentals**

The concepts are treated with the help presentation slides. A demo Web site is used to clarify the concepts. Considerable time is spent on hands-on exercises. The course times are from 9.30 up and to 16.30.

#### **Certification HTML Fundamentals**

Participants receive an official certificate HTML Fundamentals after successful completion of the course.

#### **Course HTML Fundamentals**

In the course HTML Fundamentals you will learn how to create HTML web pages. You

will learn the practice of creating and maintaining HTML pages. After an introduction on the basic architecture of the Internet in which Web Servers and Browser communicate with each other via the HTTP protocol, you learn how to create your own pages. The different elements of HTML pages, such as headers, hyperlinks, paragraphs, fonts, colors, tables, images, forms and frames are treated with demos. The most important attributes of HTML pages are also discussed. You will also learn how to use text editors and graphic editors. A final module deals with the latest version of the standard, HTML 5, in which many additions are included such as improved validation possibilities. During the course you build your own Web Site in HTML in subsequent exercises. The Web Site consists of a number of linked pages in which the most important elements of HTML work together.

Module 1 : HTML Intro	Module 2 : Basic Tags	Module 3 : Core Elements
What is HTML? HTML Versions HTML Pages HTML Elements Basic Structure HTML Page Basic HTML Document Creating HTML Pages Character Encoding Web Architecture Web Servers Browsers HTTP Protocol HTTP Protocol HTTP Request Headers HTTP Status Codes Common Status Codes	Head Tag Nested Head Tags Style Tag Common HTML Elements Heading Tag Paragraph Tag Horizontal Rules Line Breaks Formatting Tags Preformatting Ordered Lists Unordered Lists Definition Lists DOCTYPE Declaration HTML Comments	HTML Links Local Hyperlinks Global Hyperlinks Internal Hyperlinks Target Attribute HTML Images HTML Frames IFrames Tables Tables Table Attributes Empty Tag Block Tag Div Tag Span Tags Meta Tag
Module 4 : Attributes	Module 5 : HTML Forms	Module 6 : HTML 5
HTML Attributes Class Attribute ID Attribute Name Attribute Generic Attributes Body Attributes Special Attributes Event Attributes lang Attribute src Attribute alt Attribute width and height Attributes HTML Special Characters	HTML Form Structure User Input Tags Text Fields Password Fields Checkbox Fields Radio Buttons Text Areas Area Attributes Dropdown Listbox Listbox Submit Buttons Reset Buttons Hidden Fields	HTML5 Features HTML4 Pseudo Semantics HTML5 Semantic Markup Markup Differences Header Element Nav Element Footer Element Form Validation New HTML5 Input Types Email Input Type Number Input Type Date Input Type





# HTML **Fundamentals**



# **INT202: Course CSS Fundamentals**

Code: INT202

Duration: 2 days

#### Audience CSS Fundamentals Course

This course is designed for persons who wish to learn the usage of CSS for the styling of Web Pages.

#### **Prerequisites Course CSS Fundamentals**

To join this course is no specific skills or knowledge is required.

#### **Realization Training CSS Fundamentals**

The concepts are treated with the help presentation slides. A demo Web site is used to clarify the concepts. Attention is also paid to hands-on exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification CSS Fundamentals**

Participants receive an official certificate CSS Fundamentals after successful completion of the course.

#### **Course CSS Fundamentals**

This course starts with a high-level overview of Cascading Style Sheets and then



Module 1 : CSS Intro	Module 2 : CSS Specifications	Module 3 : Text and Fonts
CSS Basics What is CSS? Default browser styles Syntax Basic selector types How CSS works with HTML structure Authoring options How browsers apply styles Browser rendering differences Common Concepts	Specification Standards Brief history Current state Exploring specifications Reading specifications Checking browser support	Working with fonts Formatting text Borders Backgrounds Color
Module 4 : Layout	Module 5 : Media	Module 6 : Resources
What is the box model? Margins and padding Common units of measurement Element positioning Floats Basic layout concepts	Media types and media queries What is CSS3? Vendor prefixes What is a CSS Reset?	Resources Using frameworks and grids Popular frameworks Preprocessors Editors Online tools Additional resources





# **CSS** Fundamentals

# INT250: Course HTML5 and CSS3

#### Code: INT250

Duration: 2 days

#### Audience HTML5 and CSS3 Course

This course is intended for experienced HTML designers who want to learn how to use HTML5 and CSS3 to make modern state of the art Web sites.

#### Prerequisites Course HTML5 and CSS3

To join this course knowledge of and experience with HTML and CSS is required.

#### **Realization Training HTML5 and CSS3**

The subjects are discussed on the basis of presentation slides. The subject matter is clarified with demos. The theory is interspersed with exercises. The code is tested in different browsers. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification HTML5 and CSS3**

Participants receive an official certificate HTML5 and CSS3 after successful completion of the course.

#### **Course HTML5 and CSS3**

In the course HTML5 and CSS3 you learn to write web pages in the latest version of HTML and CSS. HTML5 and CSS3 are the dominant technologies in developing

interactive and aesthetically appealing websites. HTML still determines the content of a website (text, images) and CSS determines the style (colors, fonts and layouts), but HTML5 and CSS3 are the next step in web technologies and offer many more possibilities. The participants will become familiarized with the new HTML5 structure tags and the new input tags for form validation. They also learn to use audio and video on Web pages and also the advanced styling possibilities of CSS3 with transitions and animations are discussed. Furthermore, the possibilities of local storage, offline applications and geolocation services are discussed. Finally, web workers, messaging and web sockets are on the program. The participants learn to use push technology.

Module 1 : HTML5 Intro	Module 2 : Page Structure	Module 3 : HTML Forms
HTML5 Evolution W3C and WHATWG HTML5 Vision Compatibility Interoperability Universal Access Future of HTML5 Browser Support Browsers In Mobile Devices Feature Detection Gracefull Degradation	HTML5 Page Structure HTML5 DocType Page Encoding Pseudo Semantics HTML5 Markup New Page Elements Updated Page Elements Deprecated Elements Structure Elements New Attributes Deprecated Attributes	HTML5 Forms Gathering Input New Input Types Color Input Type Date Input Type Email Input Type New Attributes Form Validation Complex Validation Using Regular Expressions Browser Support
Module 4 : Audio and Video	Module 5 : Styling with CSS3	Module 6 : Canvas
Audio and Video Using Plugins Using Codecs Container Formats Multiple Sources Common Attributes New Audio Markup New Video Markup New Video Markup Attributes and Methods Audio and Video Events Customizing Controls	Rounded corners Drop shadows Color effects Transparency using RGBA Gradients and Rotation Scale to page elements Transitions Advanced selectors like nth-child Embed using @font-face! Using custom fonts Using text instead of images	HTML5 Canvas Using a Canvas Context and Coordinates Drawing Shapes Working with Paths Stroked Path Drawing Circles or Arcs Drawing Text and Images Working with Pixels Understanding Transforms Translation and Rotation
Module 7 : Data Storage	Module 8 : Offline Applications	Module 9 : Geolocation
Data Storage Cookies Model Pre HTML5 Solutions New Storage Options Web Storage Storage Interface Session Storage Local Storage Web SQL Storage	Offline Applications Manifest File Cache Section Network Section Fallback Section applicationCache Events Deployments And Updates error Event updateReady Event	HTML5 Geolocation Privacy Considerations Get User Location Geolocation Object getCurrentPosition watchPosition Position Object Latitude and Longitude Handling Errors
Module 10 : WebWorkers	Module 11 : Messaging	Module 12 : WebSockets





HTML

# HTML5 and CSS3



HTML5 WebWorkers JavaScript Execution Model Background Tasks WebWorker Usage Communication API Handling Errors Event Data ErrorEvent Interface HTML5 Messaging API's Same Origin Policy and Workarounds JSON with Padding Using a Proxy Cross Document Messaging Sending and Receiving Messages XMLHttpRequest Level 2 Cross Origin Resource Sharing Web Sockets Real Time Solutions Polling Architecture Long Polling Architecture Web Sockets API Web Sockets Protocol Web Sockets Handshake Advantages Web Sockets

# INT300: Course JavaScript Programming

Code: INT300

#### Duration: 3 days

#### Audience Course JavaScript Programming

The course JavaScript Programming is intended for experienced HTML designers who want to learn how to use <u>JavaScript</u> to make Web pages more interactive.

#### Prerequisites Course JavaScript Programming

To join this course knowledge of and experience with **<u>HTML</u>** is required. Prior programming knowledge is not required but beneficial in understanding the concepts.

#### **Realization Training JavaScript Programming**

The subjects are discussed on the basis of presentation slides and demos. The theory is interspersed with exercises. The code is tested in different browsers. The course times are from 9.30 up and to 16.30.

#### **Certification JavaScript Programming**

Participants receive an official certificate JavaScript Programming after successful completion of the course.

#### **Course JavaScript**

The course JavaScript Programming covers the basic principles and the use of the

JavaScript language in HTML pages. After an introduction about the various ways in

which JavaScript can be included in HTML pages and how the code is executed, the ECMA JavaScript standard is treated and the differences in handling JavaScript between browsers is discussed. Next the syntax of the JavaScript language is discussed including the variables, data types, operators and control flow constructs of JavaScript. JavaScript arrays and functions are also treated. An important aspect of JavaScript and a central element of the course is formed by the handling of events on the HTML page in JavaScript functions. The role of event functions for loading and submitting pages is treated and the events caused by mouse and keyboard interaction are discussed. Further attention is paid to the Document Object Model of HTML page and the interaction with the page through JavaScript code and the DOM API. Next validating user input on forms on the client side through code is on the course schedule. The course concludes with a discussion of object orientation in JavaScript. Follow up courses are <u>Advanced JavaScript Programming</u> and <u>TypeScript Programming</u>

Module 1 : JavaScript Intro	Module 2 : Variables and Data Types	Module 3 : Operators and Expressions
JavaScript Evolution JavaScript Characteristics JavaScript Usage ECMA Standard Dynamic HTML Script Tag Writing tot Document JavaScript Code Execution Script Files External Script Files Built-in Browser Objects Timers in JavaScript Debugging JavaScript JavaScript Tooling	JavaScript Data Types Variable Declaration Variable Initialization Identifiers Reserved Words Identifier Examples Numbers Special Values Number Usage Math Object Strings String Object Methods Booleans Objects	What is an Operator? JavaScript Operators Arithmetic Operators Comparison Operators Assignment Operators String Operators Bitwise Operator Examples Square Brackets Operator Parenthesis Operator Conditional Expression Operator Operator Precedence Expressions
Module 4 : Control Flow	Module 5 : Arrays	Module 6 : Functions
Control Structures if Statement if Examples if else Statement if else Examples Multiple Selections Nested if Statements switch case Statement Iteration Statements for Loop while Loop dowhile Loop break and continue	Creating Arrays Accessing Arrays Array Indexes Array length Processing with for Processing with for each Multidimensional Arrays Array Methods Adding Elements Deleting Elements Associative Arrays Reversing Arrays Sorting Arrays	What are Functions? Defining Functions Calling Functions Function Parameters Using Parameters Variable Scope Returning from Functions Passing Parameters Data Validation Variable Argument List apply Method Function Naming Function Best Practices
Module 7 : Events	Module 8 : DOM API	Module 9 : Validation
HTML DOM Events Register Event Handlers Event Parameter Event Handlers List Mouse Events Key Events The DOM Event Model Common DOM Events Event Object Properties DOM Level2 Event Model	Document Object Model (DOM) Browser Object Model and DOM Document Object Properties and Methods DOM Navigation DOM Manipulation DOM Node Types Node Type Properties Querying the DOM Common Element Properties	Checking Form Input Regular Expressions Meta Characters Quantifiers Character Classes String RegEx Methods Replacing RexEx Flags RegExp Object Using RegExp
Module 10 : JavaScript Objects		







JavaScript Object Orientation ECMAScript 6 Standard Classes and Object Constructors Class Members Fields and Properties Functions/Methods Access Modifiers public, private, protected Safe Usage of "this" Using JSON Namespaces

# INT303: Course Advanced JavaScript Programming

Code: INT303

Duration: 3 days

#### Audience Advanced JavaScript Programming Course

The course Advanced JavaScript Programming is designed for experienced Web developers who want to use advanced aspects of JavaScript in Web Applications.

#### Prerequisites Course Advanced JavaScript Programming

Basic knowledge of <u>JavaScript</u> and HTML is required. Prior knowledge of <u>Object</u> <u>Oriented Programming</u> is beneficial for a good understanding.

#### Realization Training Advanced JavaScript Programming

The course has a hands-on nature. The theory is treated by means of presentation slides and is interspersed with practical exercises. Demos are used to clarify the concepts. The course times are from 9.30 up and to 16.30.

#### **Certification Advanced JavaScript Programming**

Participants receive an official certificate Advanced JavaScript Programming after completion of the course.

#### Course Advanced JavaScript Programming

The course Advanced JavaScript Programming starts with a treatment of scope in JavaScript, where the difference between var, let and const variable declarations and

the term hoisting are discussed. The meaning and workings of the context dependent this keyword is also discussed here. Subsequently, attention is paid to the many new features that have made their appearance with the ECMAScript 2015 (ES6) standard, such as arrow or lambda functions, rest operators, spread operators and destructuring assignments. The changes with the ECMAScript 2016 (ES7) standard are dealt with subsequently. Then there is attention for Object Oriented programming in JavaScript that is still based on prototypes, but since ES6 is programmed with classes and objects. The JavaScript object orientation is therefore more in line with what is common in other languages. Subsequently, a number of advanced aspects of functions such as function objects, immediate invocation, callbacks, closures and recursion are discussed and attention is paid to advanced functional operations on arrays such as filter, map and reduce. After a systematic discussion of how XML and JSON data can be accessed in JavaScript, asynchronous Ajax calls with the XMLHttpRequest Object and its request processing with Promises are treated. Subsequently attention is paid to the new Web Component specifications, such as shadow and shady DOM, HTML templates and HTML Imports. Also Module Systems such as CommonJS and AMD are subjects on the course program. The last subject in the course is the use of Regular Expressions in JavaScript for a.o. input validation.

Module 1 : Scope and Context	Module 2 : ECMAScript 6 and 7	Module 3 : Classes and Objects
Scope in JavaScript var Keyword Scope in Nested Function Lexical Scope Hoisting Nested Anonymous Function let Keyword const Keyword Immutability Refresher on this this in Global Scope Lexical this Binding this apply and call Function bind Function	ECMAScript 6 Arrow Functions Object Literal Syntax Rest Parameters Spread Operator Template Strings Iteration with forof Destructuring Assignments Generators Symbols Built-in Symbols ECMAScript 7 Array.prototype.includes Exponentiation Operator Shared Array Buffer	Object Oriented Programming Objects in JavaScript Constructor Functions Object Ancestor _proto_Property Prototype Chains Prototype Keyword Composing Prototypes ES2015 Classes Constructors Static Keyword Static Methods Inheritance Overriding Methods super Keyword
Module 4 : Advanced Functions	Module 5 : Advanced Arrays	Module 6 : XML and JSON Access
Function Declarations Function Expressions Default Parameters Variable Arguments Argument Object Array Conversion Closures Immediate Invocation and IIFE's Callbacks Recursion Runtime Errors Globally Handled Errors Structured Error Handling Try and Catch	Arrays Review Iterators Arrow Functions Array.isArray() Array.from() Array.prototype.fill() Array.prototype.filter() Array.prototype.forEach() Array.prototype.forEach() Array.prototype.indexOf() Array.prototype.keys() Array.prototype.map() Array.prototype.reduce()	Document Object Model (DOM) Properties and Methods DOM Navigation DOM Manipulation DOM Node Types Querying the DOM What is JSON? JSON Data Structures JSON Object Representation JSON Array JSON with Array and Objects JSON Conversion JSON Parser JSON Object Creation





# Advanced JavaScript Programming



Ajax Asynchronous Model Creating XMLHttpRequest Sending Requests XMLHttpRequest Properties Processing the Response Accessing Remote Sites Same Origin Policy Cross Site Request Sharing JSON with Padding Promises Promises Guarantees Promise Chains Chaining after Catch ES7 Async Await

- Web Component Specifications Custom Elements Creating and Extending HTML Templates Activating Templates HTMLImports Shadow DOM Shady DOM Module Systems CommonJS Module System Asynchronous Module Definition Internal Modules External Modules Imports and Exports
- JavaScript's RegExp String Methods Regular Expression Syntax Start and End (^\$) Number of Occurrences Grouping ([]) Negation (^) Subpatterns (()) Alternatives (|) Escape Character () Backreferences Named Backreferences Form Validation Cleaning Up Form Entries

# INT400: Course Dynamic HTML

Code: INT400

Duration: 2 days

**Price:** € 999

#### Audience Dynamic HTML Course

This course is intended for person who want to learn how to make dynamic and interactive Web pages.

#### Prerequisites Course Dynamic HTML

To join this course knowledge of HTML required. Knowledge of a scripting language such as VBScript or JavaScript is recommended.

#### **Realization Training Dynamic HTML**

The concepts are treated with the help presentation slides. A demo Web site is used to clarify the concepts. Attention is also paid to hands-on exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Dynamic HTML**

Participants receive an official certificate Dynamic HTML after successful completion of the course.

#### **Course Dynamic HTML**

In this course you will learn how to create Interactive and Dynamic Web pages. You will learn XHTML, Cascading Style Sheets and Dynamic HTML capabilities. Attention is paid to design, positioning block-level and text-level elements and the in's and out's of Cascading Style Sheets. We also look at the latest developments in the World Wide Web Consortium and the differences between various browsers.

Module 1 : Advanced HTML	Module 2 : CSS	Module 3 : HTML DOM
HTML Structure DOCTYPE Standards Compliance DOCTYPE Semantic or Structural Markup Benefits of Semantic Markup Problems with Tables Headings and Paragraphs Line Breaks and Emphasis Deprecated tags HTML of XHTML Validation XML Document Structure Well Formed and Valid Documents What is XHTML XHTML document structure	What is CSS? Rendering with CSS Selectors Selector Types HTML Selectors Class Selectors SPAN and DIV as carriers ID Selectors Adding CSS Files CSS Positioning Attributes CSS Position Property CSS Visibility Property CSS Z-index Property	HTML DOM Element Access Building a DOM tree DOM levels DOM and Browser Object Model DOM Representation Node Object DOM Node Types Properties of Node Types Node properties Node methods DOM data structures NamedNodeMap interface Example Document and Tree
Module 4 : JavaScript	Module 5 : Dynamic HTML	Module 6 : Ajax
JavaScript History JavaScript Characteristics Variables JavaScript Types Numbers Strings Event Handlers onchange Event onmousedown Event Form Validation	What is Dynamic HTML DHTML Technologies script.aculo.us Use of script.aculo.us libraries Dojo Toolkit Libraries The dojo Widget Library Dojo Widget System Features Adding Widgets to Page Form Widgets	What is Ajax? Ajax term explained Classic Web Application Model Ajax Web Application Model Classic Synchronous App. Model Ajax Asynchronous App. Model Typical Ajax Interactions How Ajax Works Creating the XMLHttpRequest Object Methods Sending the Request Object Properties XMLHttpRequest readyState Listening for Response Processing the Response Sequence diagram





# **Dynamic HTML**

# INT404: Course Dojo Toolkit Programming

Code: INT404

#### Duration: 3 days

**Price:** € 1650

Module 1 : Dojo Intro	Module 2 : Dojo DOM Access	Module 3 : Dijit and Forms
What is Dojo?	DOM Manipulation	What is Dijit?
Benefits of Dojo	DOM Retrieval	Dijit Registry
Features of Dojo	DOM Creation	Dijit Attributes
JS Foundation	DOM Placement	Dijit Events
Dojo Usage	DOM Destroy	Dijit Widget Types
Dojo Architecture	Dojo Query	Menu Widgets
Dojo Base and Core	Restricting Queries	Layout Widgets
Asynchronous Module Definition	Advanced Selections	Tree Widgets
Defining Modules	NodeList Foreach	CheckBoxes and RadioButtons
Loading Modules	Connecting to Events	on Change Events
Configuring Dojo	Dojo Event Handling	NumberTextBox
Modules Loading Modules	On Method	DateTextBox
Locating Packages	Event Delegation	ValidationTextBox
Dojo Build System	Publish and Subscribe	Form Validation
Module 4 : Classes and Objects	Module 5 : Ajax Interaction	Module 6 : Routing
Classes and Objects	Ajax Term Explained	URL Modification
Encapsulation	Classic Web Application Model	Bookmarkable Pages
Prototype Based OOP	Ajax Web Application Model	dojo/hash module
Adding to Prototype	Classic Synchronous Interaction	Back Button Handling
Dojo Object Orientation	Ajax Asynchronous Interaction	Single Page App
Named Classes	XMLHttpRequest Object Methods	Topic Responses
Anonymous Classes	Sending the Request	Dojo Router
Using Mixins	Listening for Response	Route Parts
Object Sharing	Ajax in Dojo	Router Properties
Using Statics	Dojo Request	Router Callback
Single and Multiple Inheritance	Request GET and POST	Register Function
Call Superclass Methods	JSON Request	Router Responses
Constructor Chaining	JSON with Padding	Router Configuration
Module 7 : Dojo and REST	Module 8 : Dojo Stores	Module 9 : Dojo Mobile
What is REST?	Creating Stores	dojox/mobile
RESTFull Web Services	dojo/stores	Dojo Bootstrap Configuration
ID and Links	Memory Store	Dojo Mobile Template
Multiple Representations	query Method	Views and Widgets
Stateless Communications	Query Engine	Base Widgets
Content Negotation	QueryResults	FeedView
Simple Root Resource	Stateful Modeling	Settings View
Container Item Pattern	Object Data Binding	Build Profile
Map, Key, Value Pattern	DataGrid	Minimize Dependencies
Dojo Clients	Cells and Rows	Layers and Features
DojoX and Comet	Views	Building with Node.js

# **INT500: Course Ajax Programming**

Code: INT500

Duration: 3 days

#### Audience Ajax Programming Course

Web developers who want to learn how the Ajax technology can be used to develop web applications.

#### **Prerequisites Course Ajax Programming**

Knowledge of and experience with web application development especially using HTML and Javascript and also knowledge of a server environment in a language like PHP, Java or. Net.

#### **Realization Training Ajax Programming**

The subjects are discussed on the basis of presentation slides and demos. The theory is interspersed with exercises. The code is tested in different browsers. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Ajax Programming**

Participants receive an official certificate Ajax Programming after successful completion of the course.

#### **Course Ajax Programming**

The course Ajax Programming focuses on the new approach of Asynchronous

JavaScript and XML (Ajax) to develop responsive Web Applications. Ajax is not a technology standard, but describes how a number of existing technologies can be used together, including HTML, XHTML, Cascading Style Sheets, JavaScript, the Document Object Model, XML, XSLT and the XMLHttpRequest object. With the Ajax model, applications can quickly make changes to a page without having to reload. Both the creation of Ajax Requests from JavaScript on the client and the interaction with server side code is discussed in this course. The course also discusses the technologies underlying Ajax and you will learn how robust and easy to use Ajax web applications can be written. Also some Ajax frameworks are discussed and attention is paid to JavaScript Object Notation (JSON).

Module 1 : Ajax Intro	Module 2 : Ajax Processing	Module 3 : Ajax and XML
What is Ajax? Ajax Term Explained History of Ajax Classic Webapp Model Ajax Webapp Model Classic Synchronous Model Ajax Asynchronous Model Ajax Uses in the Real World Typical Ajax Interactions Drawbacks of Ajax Ajax Patterns Ajax Frameworks and links Iframes preceding Ajax	How Ajax Works Create XMLHttpRequest XMLHttpRequest Methods XMLHttpRequest Properties Sending the Request Simple Ajax Request readyState Property Dynamic HTML HTML DOM Access Read responseText Submit Data with GET Submit Data with POST Events to Start Ajax Ajax Sequence Diagram	Ajax Response Type XML DOM Load XML Document XML versus HTML DOM DOM Tree Loading in Internet Explorer Loading in FireFox DOM Navigation Navigate and Update DOM Manipulation DOM Node Types Node Type Properties Displaying Node Types Read responseXML
Module 4 : Ajax Patterns and Pitfalls	Module 5 : Ajax and JSON	Module 6 : Prototype
What are Patterns? Replacing HTML Replacing HTML Using Tabs Reading More Data Reading XML Data Avoiding Browser Cache Real Time Form Validation Auto Completion Concurrent Requests Polling AntiPattern Complex XML AntiPattern Same Origin Policy Bandwagon Effect Back Button Feedback Concurrency Open Source Javascript Unavailable	What is JSON? JavaScript Object Notation JSON Data Structures JSON Object Representation JSON Array Representation Why JSON over XML? JSON text JSON and Java Technology Send and receive JSON data Both client and server sides	What is Prototype? Prototype Framework Setting up Prototype Prototype Characteristics Prototype Shortcuts Utility functions \$(), \$F(), \$A() Form and Element classes Ajax Updater Ajax.Request Scriptaculous Library Scriptaculous Effects Scriptaculous Controls
Module 7 : jQuery	Module 8 : Dojo	Module 9 : Direct Web Remoting







What is jQuery? Ajax in jQuery Ajax Options Loading Page Fragments getJSON Ajax Limitations Ajax Global Settings Global Ajax Event Handlers Error Handling Ajax Request Completion Select what to load	What is Dojo Toolkit? Dojo Toolkit Libraries Main parts of Dojo Use Dojo from your Web Server Dojo Toolkit Package System Dojo Functions Dojo Ajax xhr wrapper functions Dojo DOM Manipulation Dojo Event System dojo.connect function Chaining function calls Publish and subscribe Dojo Widgets	What is DWR? DWR Parts DWR Operation DWR Architecture DWR Configuration DWR Converters DWR Creators Handling Ajax Calls Registering Callbacks Bean Convertors Object Convertors Utility Functions
Module 10 : Google Web Toolkit	Module 11 : ASP.NET Ajax	
What is and Why GWT? Java to Javascript compiler Javascript Obfuscation GWT Development Mode GWT Production Mode GWT Architecture JRE Emulation library GWT Project GWT Module configuration GWT UI class library Deferred Binding What is GWT RPC? Communication with Server GWT RPC Mechanism	ASP.NET Ajax ASP.NET Ajax Architecture ASP.NET Ajax Extensions Server Controls Microsoft Ajax Library Javascript Files Microsoft Ajax.js Division of Features ASP.NET AJAX Control Toolkit	

# **INT505: Course jQuery Programming**

Code: INT505

Duration: 2 days

#### Audience jQuery Course

The course jQuery Programming is targeted at experienced Web developers who want to use jQuery to implement JavaScript code in Web Applications.

#### Prerequisites Course jQuery

Knowledge and experience in developing Web Applications and with a scripting language like Javascript is required to join this course.

#### **Realization Training jQuery**

The course has a hands-on nature. The theory is treated by means of presentation slides and is interspersed with practical exercises. Demos are used to clarify the subject matter. A modern JavaScript development environment with debugging capabilities is used. The course times are from 9.30 up and to 16.30.

#### **Certification jQuery**

Participants receive an official certificate jQuery after successful completion of the course.

#### **Course jQuery**

The course jQuery Programming teaches you how to program one of the most popular

JavaScript libraries, jQuery. Attention is paid to the capabilities of jQuery and its relationship with JavaScript. After an introduction to the installation of jQuery and an overview of the library, scripts are made with the document ready function. Next it is discussed how the contents of a page can be accessed through jQuery script, where HTML tags and attributes are used for identification and styling is done with CSS. The relationship with the DOM model of the page is discussed as well. Furthermore attention is paid to the jQuery methods to filter content and also the various ways to optimize code and the use of advanced jQuery chaining statement are part of the course subjects. After discussing the manipulation of content in a page, the handling of events in jQuery is addressed. The jQuery methods that facilitate event handling are treated and attention is paid to namespacing and event delegation. The creation of effects with jQuery and the simple and advanced animation features of jQuery are treated as well. The use of animation and effects allows you to build image rotators, animated menus and tooltips, slideshows and modal windows and let Web pages come alive. Next there is attention to the Ajax functionality of jQuery for dynamically refreshing the page and for partial page updates. Both the simple \$load function to retrieve data and the more sophisticated \$ajax function are discussed and attention is paid to data transmission via JSON, JavaScript Object Notation. Finally the use of plugins in jQuery UI library and the way to write your own plugins is discussed and attention is paid to and exprised on and accessing.

Module 1 : Intro jQuery	Module 2 : DOM and Dynamic HTML	Module 3 : Selections and DOM Traversing
What is jQuery? jQuery versus Custom JavaScript jQuery versus other Libraries Getting started with jQuery Including jQuery Google Content Delivery Network Document Ready? Script Execution Order onLoad versus document.ready jQuery Function Object Where to run Scripts?	Dynamic HTML Building a HTML DOM tree DOM Representation DOM Node Types Properties of Node Types DOM Methods for Navigation DOM Node Object Methods Element Access Event Handlers CSS Style Sheets Rendering with CSS	jQuery Selection Basic Selectors Hierarchy Selectors Selection by Attribute Position Filters Form Selectors Other Selection Filters jQuery Method Chaining DOM Traversal Filter Methods Advanced Method Chaining
Module 4 : DOM Manipulation	Module 5 : CSS Styling	Module 6 : Events
DOM Manipulation Creating Elements Inserting as Child Mass Insertion Moving Elements Cloning Elements Replacing Elements Get Element Content Access Element Attributes	Browser Inconsistencies Reading CSS Properties Modifying CSS Properties Shorthand CSS Properties Removing CSS Properties CSS Classes Element Height Dimensions Element Width Dimensions Position Changes Element Positions	DOM Level 2 Event Model jQuery Event Model Binding an Event Handler Binding Shortcut Methods Event Object Unbinding Handlers One Shot Handlers Event Delegation .live versus .delegate Triggering Events
Module 7 : Effects and Animations	Module 8 : Ajax Fundamentals	Module 9 : jQuery Ajax
jQuery Built-in Effects Showing Elements Hiding Elements Fading Effects Sliding Effects Creating Animations Animation Notes Animation Queues Stopping Animations Animation Controls	Ajax Web App Model Typical Ajax Interactions Creating XMLHtpRequest XMLHttpRequest Methods Sending the Request XMLHttpRequest Properties XMLHttpRequest readyState Listening for Response Processing the Response Ajax Pitfalls	Ajax in jQuery Ajax Options Ajax Limitations Loading Page Fragments jQuery.getJSON Ajax Global Settings Ajax Events Global Ajax Event Handlers Error Handling Ajax Request Completion
Module 10 : Plugins	Module 11 : jQuery UI	Module 12 : Advanced Topics









Easing Plugin Standard Easing Functions ColorBox Plugin Using ColorBox ColorBox Options Cycle Plugin Cycle Methods Validation Plugin Validation Rules DataTable Plugin Creating Plugins jQuery UI Getting Started Themes jQuery Widgets Accordion Widget Tabs Widget Dialog Widget jQuery UI Controls Date Picker Widget Slider Widget jQuery UI Animation Best Practices for Loops Avoid Anonymous Functions Optimizing Selectors Sizzle Selector Engine jQuery Utility Methods Test Variable Type Extend Objects Avoiding Conflicts noConflict Method Queuing Animations Dequeuing Animations

# INT600: Course Rich User Interfaces using Flex

Code: INT600

Duration: 2 days

#### Audience Rich User Interfaces using Flex Course

This course is designed for experienced Web developers who want to use Adobe Flex for developing Rich User Interfaces in Web Applications.

#### Prerequisites Course Rich User Interfaces using Flex

Knowledge and experience in developing Web Applications and a scripting language like Javascript is required to join this course. This training will not focus on creating Flash applications, but on the Flex environment and Flex UI.

#### **Realization Training Rich User Interfaces using Flex**

The course is a hands-on course. The theory is treated on the basis of presentation slides and demos and is interspersed with practical exercises. The Eclipse-based development environment Flash Builder is used. The course times are from 9.30 up and to 16.30.

#### **Certification Rich User Interfaces**

Participants receive an official certificate Rich User Interfaces using Flex after successful completion of the course.

#### **Course Rich User Interfaces using Flex**





# Rich User Interfaces using Flex





This course teaches you how to use the Flash Builder development environment for rapid development of Rich Internet Applications (RIA) using Flex Framework components. You will learn how to design interactive forms where CSS stylesheets and themes are applied and where the components are given skins. You will learn to make use of the markup language MXML. Also discussed is enhancing the interactivity of the applications by using ActionScript 3.0. The data types and the object model of the ActionScript 3.0 language are covered and there is ample attention to the handling of events for the flow of the application and the validation of data. Also the subject data binding where the contents of the Flex components is linked to internal or external sources such as a Web server is addressed. Herewith the Flex Data Services such as the HTTP Service are discussed. Finally attention is paid to how robust and scalable Flex RIA applications can be written based on the MVC (Model View Controller) pattern.

Module 1 : Flex Overview	Module 2 : Flex Basics	Module 3 : Layout and Styles
Traditional Webapps Problems Rich Internet Applications RIA Benefits What is Flex? Flex Framework Flex Class Library Flex Architecture How Flex works Flex Product Line Flex Product Line Flex Compiler FlashBuilder ActionScript 3 Flash Player MXML	Minimal Flex Application Flex Containers Wine Cellar Demo UI Wine Cellar Demo Flex Components Custom Components Binding between Components Application Structure Component Properties Component Methods	Definition of Layout Layout Phases Invalidation triggers measure function updateDisplayList function Different kinds of size Size related properties Size related methods Styles Styles Styles for sub components Skins
Module 4 : ActionScript 3.0	Module 5 : MXML	Module 6 : Event Handling
What is ActionScript? Where is ActionScript used Design Goals Flash 9 Features ActionScript Basics ActionScript variables ActionScript data types Functions Classes and Objects Packages and Classes Visibility Modifiers Accessing Properties Event Model Displaylist XML handling	What is MXML? ActionScript classes to MXML tags MXML Language MXML to ActionScript MXML Visual Tags MXML Miscellaneous tags MXML Data Binding	Flex Event Model Listeners Event Listeners Examples Handling Button Clicks Inline ActionScript Best Practice Handling Events ActionScript functions Adding listeners The event object Types of Events Event bubbling Stopping the bubble
Module 7 : Data Services		
Elex Data Services		

XML Data Access Connect to data using HTTP Service Setup HTTPService Using lastResult property Handle Result using Event Object event handler in script Debugging Handling faults Fault Class Alert Class Cross Domain Access

# INT700: Course AngularJS Programming

Code: INT700

#### Duration: 2 days

Module 1 : Intro Angular JS	Module 2 : Expressions and Directives	Module 3 : Filters and Validation
What is AngularJS? AngularJS History Single Page Applications AngularJS Key Features Hello AngularJS Extending HTML MVC Pattern HTML Templates Two way Data Binding Directives Dependency Injection AngularJS Popularity Tools AngularJS Resources	Expressions JSON Data Structures JSON Object Representation JSON Array Representation JSON Arrays and Objects Built-in Directives ng-Controller Directive ng-Controller Declarations Lexical Closure ng-Bind Directive ng-BindHTML directive ngRepeat Directive ngModel Directive ngClick Directive Custom Directives	Filters Filter in Templates Currency Filter Upper and Lower Case Filter orderBy Filter Filter on Criteria Multiple Filters Date Filter JSON Filter LimitTo Filter Custom Filters Filter Anatomy Custom Reverse Filter Custom Currency Filter Using Filters in Controllers Controller Function as Filter Custom Filter in Controller Using \$filter in Controller
Module 4 : Forms and Validation	Module 5 : Routing	Module 6 : Services
AngularJS Forms Text Inputs Checkboxes Radiobuttons Select Options in HTML Select Options in Array Tracking Changes by CSS Validation Validation Styling Nested Forms Form Submission Ajax Form Submission Form Reset Touched versus Dirty	Routing AngularJS Routes Routing Configuration ng-View UI-Router Application Start Difference with ngRoute States and Services Activating States UI-Router Directives \$stateProvider \$stateParams	What are Services? Need for Services Built-In Services \$log Service \$filter Service Creating Services with Factory Injecting Services with Service Service with Provider \$http Service Using built-in Service Dependencies
Module 7 : Templates	Module 8 : Modules	Module 9 : Miscellaneous Topics
Angular Templates Static Templates ng-include \$templateCache Service Static Template in File Template Adjustment Dynamic Static Template Nested ng-include Recursive Templates Recursive JSON Structure Custom Directives Custom Element Directive templateUrl Property Variable Binding in Directive Isolating \$scope from Directive	Understanding Controllers Injected Controller in Module Avoid Global Namespace Controller in Module Controller in Separate File Two way Data Binding Inheritances Scopes \$watch Scope Inheritance Scope Inheritance Scope Inheritance Scope Event Propagation Scope Life Cycle \$watch, \$apply(), \$digest() Multiple Modules Recommended Setup	REST REST Services \$resource Service Create a Resource Resource Factory Resource Instance Animation in JavaScript Animation Performance jQuery Animations CSS Enabled Animations Animating in AngularJS Animating gRepeat ngShow and ngHide Animating ng-view

# **JAV100: Course Java Programming Fundamentals**

Code: JAV100

Duration: 5 days

Price: € 2250

#### **Audience Course Java Programming**

The course Java Programming Fundamentals is intended for developers who want to learn Java Programming and other persons who want to understand Java code.

#### Prerequisites Course Java Programming

Basic knowledge of and experience with programming is recommended. Principles of Object Oriented Programming using Java are explained. Prior knowledge of this is beneficial but not necessary for the understanding.

#### **Realization Training Java Programming**

The theory is treated on the basis of presentations and is interspersed with exercises. Demos are used to clarify the theory. The course uses Java 8. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Java Programming**

Participants receive an official certificate Java Programming Fundamentals after successful completion of the course.

The course Java Programming Fundamentals covers the fundamentals of programming in Java based on version 8 of the Java platform. In addition to the fundamentals of Java, the innovations in Java 8 such as lambdas, optionals and the

stream API are also covered in the course. By means of successive exercises in a case study, participants learn to program with the variables, data types, operators and control flow constructions of the Java language. The course then discusses object oriented concepts such as class, object, encapsulation, inheritance and polymorphism. Attention is paid to errors and exception handling and it is discussed how Java software is organized in packages. Also the split of a program into different subtasks with threads and the synchronization of these threads is the subject in the course. Parameterized types, generics, and their use in the Collection Framework are also part of the course program. Optional modules, if time permits it, are access to databases with JDBC and Java Beans. This course is a good preparation for the Java 8 OCA, Oracle Certified Associate, exam. In combination with the Advanced Java Programming course this course also prepares for the Java 8 OCP, Oracle Certified Professional, exam.

Module 1 : Basic Concepts	Module 2 : Language Syntax	Module 3 : Classes and Objects
History of Java Java Overview Language Editions Java Platform Java Community Process Libraries Java Language Security Application Types Compiling Programs Running Programs Compiler and Interpreter Application Structure Garbage Collection	Java Comments Types of Variables Primitive Data Types Block Statements Operator Precedence if else Statement for and while Loop do while Loop break and continue Arrays Enhanced for Loop Strings Formatted Output	Class Definition Encapsulation Access Modifiers Creating Objects and Constructors Fields and Methods Using Objects static Modifier this Keyword Parameter Passing Method Overloading Variable Arguments Object References final Modifier Object Destruction
Module 4 : Inheritance	Module 5 : Exception Handling	Module 6 : Packages
Single Inheritance extends Keyword Overriding Methods Hiding Attributes Calling Base Constructors Polymorphism Abstract Classes Interfaces Implementing Interfaces Default Methods Type Casting Implicit and Explicit Casting Cloning Objects	Error and Exceptions in Java Checked and Unchecked Exception Hierarchy Multiple Catch Clauses finally Clause try with Resources Exception Information Common Exceptions Throwing and Rethrowing User Defined Exceptions Chained Exceptions Stack Traces Assertions	Inside Packages Standard Java Packages java.lang Package Packages and Directories Importing Classes Wildcard Import * Full Class Names Using Packages CLASSPATH import static Packages and Visibility Packaging in JAR Runnable JARS
Module 7 : Threads	Module 8 : Synchronization	Module 9 : Special Classes
Thread Characteristics Benefits and Drawbacks Java Thread Model Thread Class Runnable interface Extending Thread Implementing Runnable Daemon Threads Thread Life Cycle States Sleeping and Yielding Control Using join and interrupt Thread Priorities	Concurrent Method Activation Synchronization Mechanisms synchronized Modifier Blocking on a Monitor Mutual Exclusion in Java synchronized Block Locking and Statics Deadlock Interthread Communication Condition Synchronization Using wait and notify while Loop and notifyall	Inner Classes Types of Inner Classes Anonymous Inner Classes Static Inner Classes Lambdas Functional Interfaces Enumerations Enum Types Declaring Enums Enums as Constant Objects Enums are Classes Enum Methods and Fields



Module 10 : Utility Classes	Module 11 : Collection Framework	Module 12 : Generics
Object Class Wrapper Classes Autoboxing and Unboxing Overriding equals Math Class Date and LocalDate Regular Expressions Scanner Class System Class Locale Class Local Class Localizing Dates and Numbers Localizing Currencies	Framework Branches Implementation Classes Collection and Map Interface Iterator Interface List Interface ArrayList and LinkedList Class Set and SortedSet Comparable Interface Comparator Interface Collections and Streams map, filter and foreach Method collect Method	Generics Explained Need and Benefits of Generics Generics in Collections Generic Characteristics Type Erasure Bounded Type Parameter Generics and Subtyping Inheritance Relationships Wildcards Bounded Wildcards Raw Types Generic Methods
Module 13 : Stream I/O	Optional Module : JDBC	Optional Module : Java Beans
I/O Basics and Classes Byte and Character Streams Standard I/O Streams Stream Types Data Sink Streams Processing Streams Buffered Streams Data Conversion Streams Serialization Serializable Classes Object Streams	JDBC Overall Architecture JDBC Drivers JDBC URL's Class forName Making Connections Executing a Statement Retrieving Results Transactions Prepared Statements Object Relational Gap Object Relational Mapping	Software Components Java Beans Properties Getters and Setters Beans Terminology Bean Component Model Bean Event Pattern Event Firing Event Class Event Listener Interface Dispatching Events

# **JAV130: Course Eclipse Introduction**

Code: JAV130

#### Duration: 1 day

Price: € 499

#### **Audience Eclipse Introduction Course**

This course is intended for developers who want to learn how to use the Eclipse IDE for software development.

#### **Prerequisites Course Eclipse Introduction**

Basic knowledge of and experience with programming is required to participate in this course.

#### **Realization Training Eclipse Introduction**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Demo's are used to clarify the concepts further. The course material is in English.

#### **Course Eclipse Introduction**

In the course Eclipse Introduction participants learn how to work with the popular Eclipse IDE for application development. Eclipse is a widely used IDE (Integrated Development Environment) for application development. Often Eclipse is used to develop Java applications but there are also plugins available for developing PHP and C++ applications. The role of views and perspectives in Eclipse is discussed and participants will learn how to configure Eclipse for a range of different tasks, such as using of a specific Java SDK or Java version and using a specific application server.





# **Eclipse Introduction**





The creation of various Eclipse projects will be discussed and participants will learn how to find their way through the multitude of settings and configuration options. Attention is also paid to debugging applications in the Eclipse Debug perspective. Furthermore the extension of Eclipse through the installation of Eclipse Plugins is part of the subject matter. Finally the integration with JUnit and the creation of JUnit tests is discussed.

Module 1 : Eclipse Intro	Module 2 : Eclipse IDE	Module 3 : Eclipse Plugin Architecture
Eclipse Project Aims Eclipse Overview Eclipse Origins Early History of Eclipse Eclipse as Java IDE Eclipse as IDE Framework Eclipse as Tools Framework Top Level Projects Eclipse Projects Eclipse Open Source Community Eclipse Eco System Eclipse Foundation	About IDE's Workspace Component Standard Widget Toolkit Eclipse on Windows XP Eclipse on MacOS Workbench Terminology Editors Views Perspectives Debug Component Ant Component Help Component Java Development Tools Java Perspective Java Editor Refactoring Eclipse Java Compiler Eclipse Java Debugger	Eclipse Layering Eclipse Plugin Architecture Eclipse Plugins Plugin Manifest Eclipse Plugin Architecture Platform Architecture Plugin Activation Plugin Fragments Plugin Install Eclipse and OSGI Import-Package Require-Bundle

#### Module 4 : Eclipse JUnit

Test Driven Development Unit Testing What is and why JUnit? JUnit Features JUnit Based Test Code JUnit mechanics Simple Test Simple Test Assert Statements Fixtures Test Suites Test Runner JUnit classes

# **JAV175: Course Java Certification**

Code: JAV175

Duration: 2 days

#### **Audience Course Java Certification**

The course Java Certification is intended for experienced Java developers who want to prepare themselves for the Java 8 programmer exams OCA (Oracle Certified Assciate) and OCP (Oracle Certified Professional).

#### **Prerequisites Course Java Certification**

To participate in this course knowledge of and ample experience with object oriented concepts and Java programming is required.

#### **Realization Training Java Certification**

The course is an exam training in which, apart from theory and demos, various test exams with test questions for the exams are discussed. The theory is interspersed with practical exercises. The course material is in English. The course times are from 9.30 am to 16.30 pm.

#### **Official Certificate Java**

After successful completion of the course attendants receive an official certificate of participation in the Java Certification course.

#### **Course Java Programmer Certification**

In the course Java Certification participants are prepared for the Java OCA (Oracle Certified Associate) and OCP (Oracle Certified Professional) exam. The course is an exam training in which the central focus is on the questions that can be asked on these exams. On the basis of test questions and test exams, the subjects that are part of the exams are treated. Special attention is paid to topics that are often considered difficult such as concurrency and synchronization. As well as features that have been added in later versions of Java such as lambda's and streams. Also discussed are generics, collection classes, database access with JDBC and new I/O. Subjects like declarations and access control, object orientation, assignments and operators, flow control, exceptions and assertions, strings, I/O formatting and parsing, inner classes are also on the program. Depending on the interest of the participants certain components can be treated with more depth if so desired.

Module 1 : Concurrency	Module 2 : Lambda's and Functional Interfaces	Module 3 : Generics and Collections
Concurrency Package Task Scheduling Framework Executor Interface ExecutorService Callables and Futures ScheduledExecutorService Synchronizers Semaphores and Exchanger CountdownLatch CyclicBarrier Concurrent Collections BlockingQueue Interface Lock Interface Reentrant Locks Atomic Variables	Passing Functionality Lambda Expressions Lambda Syntax Lambda Variable Access Lambda Scoping Rules Functional Interfaces Predicate Interface Consumer Interface Supplier Interface Function Interface UnaryOperator Interface BinaryOperator Interface Method References @FunctionalInterface annotation User Defined Functional Interfaces	What are Generics? Type Erasure and Raw Types Generics and Subtyping Bounded Type Parameters Wildcards Generics in Collections ArrayList and LinkedList TreeSet and Hash Set HashMap and TreeMap ArrayDeque objects Comparable and Comparator Interface Collections Streams and Filters Iteration using forEach Filtering using Lambda's Stream Interface and Stream Pipeline
Module 4 : Database Access with JDBC	Module 5 : Streams	Module 6 : New IO
JDBC Architecture JDBC Drivers and URL's Database Connections Executing Statements Querying Databases Update Statements Retrieving Results Handling Errors Prepared Statements Database Metadata Transactions Commit and Rollback Rowset Interfaces	What are Streams? Lazy Evaluation and Parallelization Core Stream Methods forEach, Map and Filter findFirst and findAny toArray and collect Optional Class Limiting Stream Size allMatch and anyMatch Number Specialized Streams Reduction Operations Parallel and Infinite Streams collect and flatMap Method	What is NIO? Synchronous I/O Processing Asynchronous I/O Processing Working with Buffers IO Channels Selectable Channels Selectors Selectors Selection Keys Character Sets Using Path Class Directory Traversing PathMatcher class Using WatchService





# Java Programmer Certification

# JAV200: Course Java EE Web Development

Code: JAV200

Duration: 4 days

#### Audience Course Java EE Web Development

The course Java EE Web Development is intended for developers who want to build Java Web applications with Java EE components like Servlets, JSP's custom tags, filter and event listeners.

#### Prerequisites Java EE Web Development

To participate in this course experience with Java programming and object orientation is required.

#### **Realization Training Java EE Web Development**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Demos are used to clarify the theory. All topics asked at the Java Web Component Developer Exam are discussed. The course material is in English.

#### **Official Certificate Java EE Web Development**

Participants receive an official certificate Java EE Web Development after successful completion of the course.

#### **Course Java EE Web Development**

The course Java EE Web Development covers how servlets and JSPs are constructed

and used in Java EE Web Applications. First the main methods of servlets are on the agenda, like the ones used in servlet initialization, reading HTTP request data and writing HTTP response data. The different scopes of Web Applications such as session scope and application scope are explained. Next the basic syntax of JSP's scriptlets, expressions and declarations are covered and the different JSP page directives are treated. A central element in the course is the MVC design pattern for Java Web Applications where servlets act as controller for program logic and JSP's as view for output data. In discussing this pattern, the RequestDispatcher interface is treated as well as the ways attributes can be stored and retrieved in different scopes. Ample attention is paid to techniques to separate Java code from the JSP view. In particular the use of Java beans and custom tags for this purpose is discussed. Also the role of deployment descriptors is part of the subject matter and security in Web applications, filters and the event framework are addressed.

Module 1 : Introduction	Module 2 : Servlets	Module 3 : Scopes and Sessions
Java EE Standard Java EE Servers What is a Servlet? Possible Servlet Tasks What is a Java Server Page(JSP)? Problems with Servlets and JSP Classic MVC Pattern Model 2 Architecture Structure of a Web Application Registering a Web Application Defining Custom URL's WAR Files Handling Relative URL's	Servlet Interface Reading Initialization Parameters Concurrent Access ServletContext Interface Destroying Servlets HTTP Requests and Responses HttpServlets service, doGet and doPost Reading Form Data HTTP Request Headers Populating the Response HTTP Status Codes HTTP Response Headers	Sharing Data with Scope Objects ServletContext Application Scope Request Scope Page Scope Session Scope Session Tracking Mechanisms Sending and Reading Cookies Session Tracking with Cookies URL Rewriting Hidden Form Fields HttpSession methods Session tracking API
Module 4 : JSP's	Module 5 : JSP Directives	Module 6 : JSP Standard Actions
The Need for JSP Types of Scripting Elements Benefits of JSP Translation and Request Time Setting up your environment Expressions Predefined Variables Scriptlets Declarations jsplnit and jspDestroy	What are JSP directives JSP page Directive import Attribute contenttype Attribute session Attribute buffer and autoflush Attributes extends Attribute errorPage and isErrorPage include Directive taglib Directive	jsp:include jsp:plugin, jsp:param and jsp:params What are Beans? Basic use of Beans in JSP Accessing Bean Properties Setting Bean Properties Explicitly Association with Input Parameters Sharing Beans Values of Scope Attribute Conditional Bean Operations
Module 7 : MVC Architecture	Module 8 : Expression Language	Module 9 : Custom Tags
Why combine Servlets and JSP? MVC Approach Implementing MVC Dispatching requests Storing Data in Servlet Request Storing data in ServletContext Forwarding Requests Including Requests	Advantages Expression Language Activating Expression Language Invoking the EL Common EL Problem Referencing Scoped Variables Accessing Bean Properties Nested Beans Using EL Operators Conditional Evaluation	What are Tag Libraries? Custom Tag Features Simple Tag Handler Simple Tag Library Descriptor Accessing Custom Tags from JSP Attributes and Body Content JSP-based Tags Tags manipulating Body Complex Objects for Attributes
Module 10 : Security	Module 11 : Filters	Module 12: Event Listeners





# Java EE Web Development





Major Security Concerns Declarative Security Programmatic Security Form-based Authentication Deployment Descriptor settings BASIC Authentication Combining Security Mechanisms Pure Programmatic Security Programmatic Security with SSL What are Filters? Filter Interface Creating Filters doFilter method Servlet Filter Related Classes Filters in a Chain Filter Mapping Accessing the Servlet Context Filter Initialization Parameters Life-Cycle Events Framework Available Listeners Implementation Strategy Reason for listeners ServletContext Listeners ServletContextAttributeListeners HttpSession Listeners Session Creation Events SessionAttribute Listeners

# JAV250: Course Java EE Web Development with JSF

Code: JAV250

Duration: 4 days

#### Audience Course Java EE Web Development with JSF

The course Java EE Web Development with JSF is intended for developers who want to build Java Web applications with the standard JSF 2.x Framework.

#### Prerequisites Java EE Web Development with JSF

To participate in this course experience with Java programming and object orientation is required. Prior knowledge of Java EE Web Applications is beneficial for a proper understanding.

#### Realization Training Java EE Web Development with JSF

In this hands-on course the theory is treated on the basis of presentation slides and interchanged with exercises. Demos are used to clarify the theory. The course material is in English.

#### **Official Certificate JSF**

Participants receive an official certificate Java EE Web Development with JSF after successful completion of the course.

#### Course Java EE Web Development with JSF

The course Java EE Web development with JSF course covers how modern state of

the art Java EE Web Applications are built with the standard Java Server Faces (JSF 2.x) framework. The course starts with the foundations of the JSF Framework which is built on top of the Java Web Programming model uses the MVC pattern with a Front Controller as central element. The architecture of the JSF Framework which uses a statefull component model on the server is discussed. Next attention is paid to the role of the Faces Servlet and the different phases of the JSF Request Life Cycle. The role of the central configuration file faces-config.rml is explained and the navigation rules in JSF and the role of the managed beans are treated. Also the structure of a JSF page using Facelets, the use of expression language and the usage of the different UI components and custom tag libraries is discussed. The handling of events coming from the UI elements is also covered. Next attention is paid to the validation and conversion of input from the User Interface using JSF built-in or custom validators and converters. Also it is discussed how the modern Ajax asynchronous request model can cooperate with JSF. Finally attention is paid to the creation of custom components, facelet templates and view parameters.

Module 1 : JSF Intro	Module 2 : JSF Basics	Module 3 : Managed Beans
JSF Background Servlet and JSP Problems Classic MVC Pattern Model 2 Architecture Struts Framework JSF Framework JSF Feature Overview Server Side UI Components JSF Application Architecture Request Processing Phases JSF Implementation JSF API Common JSF Problems	Choose JSF Implementation Deployment Descriptor PROJECT_STAGE Development Faces Configuration File Facelets Page Structure Managed Beans @ManagedBean Annotation Default Navigation Facelet Default Navigation @FacesComponent Handling Request Parameters Facelet Input Form Selecting Result Pages	Java Beans Bean properties Accessor Usage Placeholder for Results Dual Role Bean Properties Textfields and Checkboxes Prepopulating Input Fields Naming Beans Bean Scopes Application and Session Scope Raw Request and Response Dependency Injection ManagedProperty
Module 4 : Navigation	Module 5 : Properties	Module 6 : Expression Language
Static Navigation Navigation Rule Syntax Explicit Mapping Considerations Dynamic Navigation Dynamic Navigation Logic Explicit Dynamic Navigation Rule Configuring Managed Beans Sharing Result Pages Using Wildcards Omitting from-outcome Conditional Navigation Rules Common Navigation Problems	Properties Using Property Files Using Subfolders Resources Bundles Facelet using Properties Parameterized Messages Parameterizing Display Using f:param Using Parameterized Properties Internationalized Messages Localizing Strings Using Localized Messages	JSF versus JSP EL Outputting Bean Properties Nested Bean Properties Submitting Bean Properties Getter versus Setter Calls Accessing Collections Using Square Brackets Implicit Objects EL Operators Conditional Text Rendered Attribute Method Arguments
Module 7 : Event Handling	Module 8 : Validation	Module 9 : Ajax
User Interface Events Event Handlers versus Ajax Types of Event Listeners ActionListener in Facelets Implementing ActionListener Change CSS Style Change Style Input Form Change Style Bean Bean for Action Controller Locale from User Settings Switch Locale ValueChangedListener	Validation Approaches Manual Validation Standard Error System Displaying Error Messages Manual Validation in Bean Action Controller Method Implicit Automatic Validation Precedence Validity Tests Explicit Automatic Validation Format Validator Tags Conversion versus Validation Custom Validator Methods	Ajax in JSF General Form f.ajax Ajax Facelets Pages Ajax Controller Methods render Attribute Facelet Non-Ajax Version Facelet Ajax Version execute Attribute Facelets Code execute event Attribute Default Events onevent Attribute
Module 10 : Data Tables	Module 11 : Composite Components	Module 12 : Facelet Templating



# Java EE Web Development with JSF 2.x





Handling Variable-Length Data h:dataTable Using h:dataTable h:Column Headers and Captions StyleSheets Ajax-enabled tables Tables conditional values Looping with ui:repeat Conditional Text Composite Components Component Definition Using Components Components with Attributes Using Composite Attribute Component Structure Using ui:repeat Relative URL's Nested Components Component Files Page Templating Using Templates Namespaces Template Files Template File Details Using ui:include Handling Relative URL's View Params JSF page use f.viewParam Capture Request Parameters Using POST-redirect-GET

# JAV300: Course Java EE EJB Business Components

Code: JAV300

Duration: 3 days

#### Audience Course Java EE EJB Business Components

Java developers that want to learn how to develop of Java EE 6 EJB business components.

#### Prerequisites Java EE EJB Business Components

To participate in this course knowledge and experience with Java programming is required. Knowledge of Java Web development is beneficial for a proper understanding.

#### **Realization Training Java EE EJB Business Components**

The theory is explained using presentation slides and is interspersed with practical exercises. Demos are used to clarify the theory. All topics that are asked in the Java EE 6 Enterprise JavaBeans Developer Certified Expert Exam (1Z0-895) are discussed. The course material is in English.

#### **Course Java EE EJB Business Components**

This course focuses on Enterprise JavaBeans, as specified in the EJB 3.x specification as part of the Java EE 6. Attention is paid to the different types of Enterprise Beans like Session Beans, Message Driven Beans and Singleton Beans. The Java Naming and Directory Interface (JNDI) is discussed and the simplification of how beans can be located and instantiated through injection. Attention is also payed

to the lifecycle of the different types of beans and to concurrency issues. The important role that annotations play in Java EE and EJB's is discussed as well. Annotations do their work in many places such as in establishing links with resources and the realization of persistence. Also the use of the EJB Query Language and the implementation of Object Relational Mapping in EJB 3.x is part of the course schedule. Also the use of transactions in a Java EE 6 environment is considered. Attention is paid to the Java EE security architecture and the various authentication and authorization strategies. JMS and its use in combination with Message Driven Beans is also on the course program. And finally the focus is on best practices and design patterns in EJB technology.

Module 1 : Java EE Intro	Module 2 : Enterprise Java Beans (EJB)	Module 3 : JNDI and Injection
Java Editions Java EE Enterprise Applications Java EE Servers Web Components EJB Components JavaBean components Web Services Java EE API's EAR Files Deployment Descriptors Annotations Java EE 6	EJB features Distributed Objects Session Beans Statefull and Stateless Architecture of an EJB EJB Object at work Client Access Remote versus Local Clients Web Service Clients Singleton Beans Singleton Beans Singleton Concurrency Access Concurrency management strategy Session faÇade EJB Programming Model Life Cycle Session Beans Message Driven beans Life Cycle Message Driven Beans	Locate resources with JNDI JNDI Architecture JNDI InitialContext JNDI Naming Context EJB Environment Standard JNDI SubContexts Dependency Injection DataSource Injection Injection of EJB References Example EJB Injection
Module 4 : Java Persistence API	Module 5 : Entity Callbacks and Listeners	Module 6 : Session Beans
Persistence API Entity Classes Entity Manager Persistence Context Entity Identity Entity Managers Entity Lifecycle Entity Relationships Persisting Objects Removing Objects Merging Objects Managing Identity	Life Cycle Callback methods Entity Listeners Life Cycle Callback Rules Signature Life Cycle Callbacks Signature Entity Listeners @PrePersist and @PostPersist @PreRemove and @PostRemove @PreUpdate and @PostLoad Multiple Invocation Callbacks Invocation Order	Session Beans Overview Stateless Session Beans Lifecycle Stateless Session Beans Event callbacks Asynchronous communication Singleton session bean Singleton concurrency access Stateful Session Beans Servicing Remove Passivating and Activating Cart Session Bean Remote Business Interface Cart Session Bean Class Life Cycle Callback Methods Calling Business Methods Remove method
Module 7 : Message Driven Beans	Module 8 : Timer Service	Module 9 : Interceptors





# Java EE Business Component Development with EJB's





Motivation Message Driven Beans Messaging Characteristics Message Oriented Middleware Publish and Subscribe Point tot Point What is JMS? Message types Creating and Receiving Messages Integrating JMS with EJB Message Driven Beans MessageListeners onMessage method	Timer Service Scheduling Timers Creating Timers Timeout method rules Canceling and Saving Timers Timer Usages Timer Service interface Timer interface Timer handle interface	Interceptor Interceptor Classes @AroundInvoke Example Interceptor Interceptor Lifecycle Interceptor Types Default Interceptors Exclude Interceptors
Module 10 : Transactions	Module 11 : Security	Module 12 : EJB and Web Services
Transactions Demarcating Boundaries Container Managed Bean Managed Client Managed Transaction Attributes SessionSynchronization Before Completion After Completion	Java EE security architecture Java EE and EJB Security Security Overview Programmatic Security Java EE authorization strategies Declarative Security Security Functions Using Programmatic Security Method Permissions DeclareRoles ejb-role-ref role-link	What is a Web Service? RPC Example Structure SOAP Message What is WSDL Stateless Session Bean Endpoint JAX-WS Publishing an Endpoint Web Service Annotations
Module 13 : EJB Best Practices	Module 14 : Clustering	Optional Appendix : Annotations
Define best practices Benefits of EJB best practices Patterns to Java EE application design Implements effective exception handling EJB Design Patterns	What is clustering? Thin Client Clustering JBoss all configuration Clustering Stack What is JGroups What is JCache HA-JNDI HA-Smart Proxies SLSBs SFSBs HTTP session replication Entity beans EJB3 POJO Clustering	What are annotations? Pre-Java 5 annotations Why annotations Defining annotation types Using annotations Kinds of annotations Marker annotations Single value annotations Normal annotations Meta-Annotations
Optional Appendix : JMX		
What is JMX? JMX API IMX Goal		

JMX API JMX Goal Managed Beans MBean flavors JMX Architecture Java SE 5.0 Mbeans Naming MBeans MBean Server Registering Mbeans Manipulating MBeans Notifications Notifications Notification Listeners

# JAV303: Course Java ME Mobile Programming

Code: JAV303

#### Duration: 3 days

#### Audience Java ME Mobile Programming Course

The course is designed for Java developers who want to learn how to develop mobile applications in Java.

#### Prerequisites Course Java ME Mobile Programming

To join this course programming experience with Java and basic knowledge of embedded systems and devices is required.

#### **Realization Training Java ME Mobile Programming**

The concepts are treated on the basis of presentation slides. Illustrative demos on emulated devices are used to clarify the concepts. There is ample opportunity to practice and theory and exercise are interspersed. The course material is in English.

#### **Course Java ME Mobile Programming**

In this course the Connected Limited Device Configuration (CLDC) as part of the Java Micro Edition (Java ME) is addressed. The course focuses on how the CLDC provides a Java run-time environment for small devices with limited resources such as mobile phones, personal digital assistants (PDAs) and POS (Point of Sales) terminals. Attention is paid to the K virtual machine (KVM) and the development of applications for mobile phones and PDA's with the Mobile Information Device Profile (MIDP). Also the creation of user interfaces for mobile phones, data access on mobile phones and





#### Java ME Mobile Programming





Internet connections are part of the course schedule. The Connected Device Configuration (CDC) as part of the Java Micro Edition (Java ME) is discussed as well. Here the CVM virtual machine, the Foundation Profile and Personal Profiles are addressed. Finally, attention is paid to various optional packages.

Module 1 : Java ME	Module 2 : MIDP Overview	Module 3 : MIDP GUI Programming
Java ME Back to the roots Place of Java ME Java Community Process Java 2 Platform Architecture Relation to other technologies Organization of Java ME Java ME Architecture Configuration and Profiles CLDC Configuration CDC Configuration MIDP Profile IMP Profile CDC Profiles Optional Packages JWTI specification	Mobile Information Device Profile MIDIets MIDIet Life Cycle States MIDIet Life Cycle Methods MIDIet Build Cycle Preverification MIDIet Suites Packaging the application JAD and JAR Manifest MIDIet Suite Execution Environment	MIDP GUI Programming MIDP UI Methodology MIDP UI Classes High and Low level API MIDP GUI Programming Model Display and Displayable Screen and Canvas Alert, List, TextBox and Form Form Items Choice Groups Tickers Gauges Guidelines for MID UI Programming
Module 4 : MIDP Event Handling	Module 5 : MIDP Data Access	Module 6 : MIDP Networking
User interaction Callbacks Screen Navigation with Commands Command Placement Handling High Level Commands	Persistent Storage Record Management System Characteristics of the Record Store RMS class and interfaces RMS Exceptions Record Stores Programming RMS Record Store States Reading and Updating Records Deleting and Closing Sorting and Searching Getting RecordID Metadata on Record Store	CLDC Generic Connection framework Networking interface Connection Interface Hierarchy Connection Examples HTTP Connection interface Connection Setup Connection State Sending a HTTP GET Request Sending a HTTP POST Request Added features in MIDP 2.0
Module 7 : MIDP 2.0	Module 8 : Mobile Media API (MMA)	Module 9 : Wireless Messaging API (WMA)
MIDP 2.0 Specification Goals Media API Play Sounds Enhanced Forms Enhanced networking Secure networking Code Signing and Permissions Enhanced Game API Push functionality OTA Provisioning	Mobile Media API Overview of MMAPI Multi Media Processing The MMAPI Architecture MMAPI Packages Tone Generation MP3 Playback Video Playback Camera support	Wireless Messaging API WMA Packages WMA 2.0 Message Types Message and Sub interfaces Structure MMS Multipart Message WMA Connections Connection relation to GCF Creating Connections SMS Send SMS Receive
Module 10 : Game API in MIDP 2.0	Module 11 : BlueTooth API	Module 12 : Other J2ME optional packages

Game API Overview The Game Package Example Game Game loop MIDP 1.0 Game Loop MIDP 2.0 Game Loop Game Canvas Sprites Sprite Animation and Manipulation Managing Tiles and Layers User Input Collision Detection Concurrency What is Bluetooth? Frequency Hopping Java API's for Bluetooth Bluetooth Network Topology Establishing Bluetooth connection Bluetooth in Java ME Device discovery Service discovery Communication Java ME Platform Specification Web Services API for Java ME Location API for Java ME Security and Trust Services API Session Initiation Protocol (SIP) RMI for Java ME JDBC for Java ME PDA Optional Packages Event Tracking API for Java ME SVG (Scalable Vector Graphics) API 3D Graphics API Mobile Internationalization Content Handler API

# JAV333: Course Development with Maven

Code: JAV333

Duration: 2 days

Built by:

#### Audience Using Maven Course

This course is intended for developers who use Maven for dependency management and for the automatic building and deployment of projects.

#### **Prerequisites Course Using Maven**

To participate in this course knowledge of and experience with Java and XML is required.

#### **Realization Training Using Maven**

The theory is discussed on the basis of the presentation slides and is interspersed with exercises. Demo projects in Maven are used to clarify the concepts. The course material is in English. The course focuses on Maven version 3.

#### **Certification Development with Maven**

Participants receive an official certificate Development with Maven after successful completion of the course.

#### **Course Using Maven**

In this course participants learn the skills and acquire the knowledge needed to use

Maven as an automated build and dependency management tool. First an overview of the problems in project and dependency management is given and the operation of Maven and role of Maven repositories is explained. Next it is discussed how Maven relates to the automatic build tool Ant. Then the directory structure of Maven projects and the standard life cycle is addressed. Maven projects can be created on the basis of predefined archetypes in which a given project structure is already present from the start. Attention is paid to a number of archetypes and to the role of goals and Maven plugins. Also the Project Object Model (POM) with pom.xml is discussed. The significance of the main entries in it such as Group, Artifact, and Version, are treated and attention is also paid to a complex structure with multiple pom files and pom inheritance. Finally the role of Maven when performing tests, continuous integration and release management with Bamboo are examined.

Module 1 : Maven Intro	Module 2 : Core Concepts	Module 3 : Build Lifecycle
Java Build Tools Intro	What is Maven?	What is Build Lifecycle?
Desired Features	Why Maven?	Standard Lifecycles
Ant + Ivy	Convention over Configuration	Key Lifecycle Phases
Build.xml	Maven Directory Structure	Build Phases and Goals
Build File with Ivy	Project Object Model	Clean Lifecycle
Ivy Dependency Management	Maven Project Coordinates	Default or Built Lifecycle
Maven	POM Structure	Default Lifecycle Phases
Build Lifecycle	POM Sections	Site Generation and Reporting
pom.xml	Plugins	Site Lifecycle
Gradle	Archetypes	Site Website
Results Matrix	Catalog File	Customizing the Lifecycle
Tools Comparison	Dependencies	Package-specific Lifecycles
Module 4 : Profiles	Module 5 : Plugins and Goals	Module 6 : Archetypes
Environment variables	What are Maven Plugins?	What is an Archetype?
User-defined properties	Plugin Types	Different Archetypes
Filtering Resources	Goals and Plugins	Archetype Generate Command
Build Profiles	Key Plugin Concepts	Maven Archetype Archetype
What is a Build Profile?	Maven Antrun Plugin	Maven Archetype WebApp
Project Configuration with Profiles	Maven Compiler Plugin	Simple J2EE Project
Profile Activation	Exec Maven Plugin	Maven Archetype Simple Site
Explicit Profile Activation	Jetty Maven Plugin	Creating Archetypes
Activation via Maven Settings	Eclipse Maven Integration	Maven Archetype Plugin
Activation via Environment Variables	Maven Checkstyle Plugin	Archetype Descriptor
Activation via Operating System	Findbugs Maven Plugin	Prototype POM
Activation via Files	Maven PMD Plugin	Prototype Files
Module 7 : Repositories	Module 8 : Dependency Management	Module 9 : Build and Deployment Automation
What is a Maven Repository?	What is Dependency Management?	Handling Rapid Changes
Enterprise Repositories	Searching Dependencies	What is a Snapshot?
Local Repository	Transitive Dependencies	Snapshot Dependency
Central Repository	Dependency Terminology	Build Automation
Repositories in Super POM	Dependency Scope	Using Snapshots
Remote Repository	Optional Dependencies	Release Management
Maven Search Sequence	Version Ranges	The Maven Release Plugin
Plugin Repositories	Project Versions	Developer Release Workflow
Repository Management	Visualizing Dependencies	Integration with Source Control
Deploying to Nexus with Maven	Dependency Conflicts	Continuous Integration
Performing a Staged Release	Excluding Transitive Dependencies	Deployment Automation



Naven



# **Using Maven**

Built by

# JAV350: Course Java for Testers

Code: JAV350

Duration: 3 days

#### **Audience Course Java for Testers**

The course Java for Testers is intended for experienced testers that want to learn how to use Java, JUnit and Cucumber for writing test scripts.

#### **Prerequisites Course Java for Testers**

Experience with testing is required to join this course. Programming experience is beneficial for the understanding of the concepts but is not required.

#### **Realization Course Java**

The course has a hands-on nature. The theory is treated on the basis of presentation slides and is interspersed with practical exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

#### **Certification Course Java Testers**

Participants receive an official certificate Java for Testers after successful completion of the course.

#### **Course Java for Testers**

In the course Java for Testers you will learn the basic syntax of **Java** and how to use it when writing JUnit test scripts and Cucumber scenarios. Attention is paid to the

various data types in Java and the flow control constructions. Also the use of classes and objects with methods, parameter passing and private and public data is discussed. And exception handling in Java and the Java collection framework are treated as well. Attention is also paid to the design of JUnit tests, the structure of JUnit tests, the annotations and assert statements used, the execution of JUnit tests in an Eclipse environment as well as the reporting on the outcome of the tests. Finally the Cucumber Framework will be discussed and the mini language Gherkin and the integration with JUnit. The demos and exercises in the course are done in a JUnit environment and are therefore directly applicable when writing test scripts. Attention is also paid to the automation of tests in the context of setting up continuous integration. The projects are built in the dependency management tool Maven. The course provides a good basis for participating in the course <u>Web</u> Testing with Selenium.

|--|

## **Java for Testers**



Module 1 : Basic Concepts	Module 2 : Language Syntax	Module 3 : Classes and Objects
Java Overview	Java Comments	Class Definition
Java Editions	Variables	Encapsulation
Java Platform	Types of Variables	Access Modifiers
Java Community Process	Primitive Data Types	Constructors
Java Language	Block Statements	Creating Objects
Compiling Java Programs	Operator Precedence	Fields and Methods
Running Java Programs	Flow Control	Using Objects
Compiler and Interpreter	if else Statements	static Modifier
Application Structure	switch Statement	this Keyword
Packages	for and while Loop	Parameter Passing
Jar files	break and continue	Method Overloading
Classpath	Arrays	Object References
Java Libraries	Enhanced for Loop	final Modifier
Maven	Strings	Object Destruction
Module 4 : Exception Handling	Module 5 : Collections	Module 6 : JUnit and Cucumber
Error Conditions	Collection Framework	What is JUnit?
Exceptions in Java	Framework Branches	Annotations
Exception Handling Syntax	Implementation Classes	Test Cases
Exception Hierarchy	Legacy Collections	Assert Statements
Multiple Catch Clauses	Collection Interface	Fixtures
finally Clause	Iterator Interface	Test Suites
Exception Information	Concrete Collections	Testing for Exceptions
Predefined Exceptions	List Interface	What is Cucumber?
Common Exceptions	ArrayList Class	User Stories
Throwing Exceptions	LinkedList Class	Scenario's
User Defined Exceptions	Map Interface	Feature Files
Chained Exceptions	Ontional Methods	Gherkin Keywords

# JAV400: Course Java EE Web Services

Code: JAV400

#### Duration: 4 days

Price: € 1950

#### Audience Course Java EE Web Services

The course Java EE Web Services is aimed at experienced Java developers who want to develop Web Services in a Java EE environment.

#### **Prerequisites Java EE Web Services**

To join this course, knowledge of and experience with programming in Java and Java EE Web Applications is required.

#### **Realization Training Java EE Web Services**

This course has a hands-on character. The theory is covered on the basis of presentation slides and is interspersed with practical exercises. Demos are used to clarify the theory. The course material is in English.

#### **Certificate Java EE Web Services**

Participants receive an official certificate Java EE Web Services after successful completion of the course.

#### **Course Java EE Web Services**

In the course Java EE Web Services you will learn what Web Services are, how you

can create Web Services in Java and how they can be accessed from Java and other

platforms. The course covers both SOAP Web services and REST Web services. The various standards on which Web Services are based are treated, such as XML Schema, SOAP, WSDL and HTTP for REST Services. In particular, attention is paid to the various Java APIs for Web Services such as JAX-WS 2.x, SAAJ (SOAP with Attachment API), JAXB (Java API for XML Binding) and JAX-RS. Attention is also paid to what you should look out for to ensure that Web Services are interoperable between different platforms such as Java and .NET. Also the architecture of modern Single Page Applications with a REST service backend is discussed. And finally, the various mechanisms and standards for the security of Web Services are discussed. This course covers the topics that are asked on the Oracle Java EE Web Services exam.

Module 1 : Web Services Intro	Module 2 : SOAP	Module 3 : SAAJ
What are Web Services? Distributed Applications Evolution Role of interface Interoperability Web Service Types Web Services Stack SOAP Web Services REST Web Services REST Web Services Micro Services Document Style Web Services Service Oriented Architecture	What is SOAP? SOAP Characteristics SOAP Messages SOAP Body SOAP Headers SOAP Namespaces SOAP Faults SOAP Version differences SOAP Messages as payload Message Exchange Patterns SOAP Intermediaries actor and mustUnderstand attribute	What is SAAJ? SOAP message structure SOAP Message Parts SOAP Part Attachment Parts SAAJ and DOM SAAJ Class Hierarchy SAAJ Connections Creating a Message Adding Message Elements Sending and Receiving Accessing SOAP Body
Module 4 : XML-Schema	Module 5 : JAXB	Module 6 : WSDL
Why XML-Schema? Well formed and valid documents What XML-Schema's? Markup Languages XML Schema Advantages XML Schema design models Classic Use of Schema's XML Namespaces Simple and Complex types XML Schema Data Types User Defined Data Types Derivation by Restriction Derivation by Extension	XML Processing Options Java API for XML Binding JAXB Architecture JAXB Binding Life Cycle Role of Binding Compiler XML tot Java Mapping Rules Mapping of XML Schema Types JAXB API JAXB Annotations Types Binding Elements and Attributes Named Complex Types Customized Mappings Adapter Classes	Web Service Description Language WSDL Usage WSDL and Code Generation WSDL in Web Service stack WSDL Namespaces WSDL Structure WSDL Elements Types and Messages PortType and Operations WSDL Bindings Service Element SOAP Messages Modes WSDL 2.0
Module 7 : JAX-WS	Module 8 : Message Handlers	Module 9 : Asynchronous Calling
What is JAX-WS? JAX-WS Design Goals JAX-WS Runtime System JAX-WS Tools Service Endpoint Models JAX-WS Servlet Endpoint Enterprise Java Beans Endpoints JAX-WS Architecture Client Side Programming Models Dynamic Proxy Invocation Model Dispatch Invocation Model	Message Handlers Characteristics JAX-WS Handler Types SOAP Message Handlers Logical handlers Call Chain Inbound Messages Outbound Messages Processing the Payload Handler Chains Configuring Handler Chains	Web Service Invocations Supporting Asynchrony Client Side Fire and Forget Client Side Polling Client Side Callbacks Enabling Asynchronous calls Callback Handler Calling Asynchronously Asynchronous Web Services Asynchronous Web Service Model Using WS-Addressing
Module 10 : REST Services	Module 11 : Interoperability	Module 12 : Web Service Security





## Java EE Web Services




Representational State Transfer REST Web Service Principles ID and Links REST Services in Java JAX-RS and Rest Jersey Content Negotation Embedded Path Parameters @Path and @Param Multiple Representations @Produces and @Consumes Common REST Patterns Resources URI Access XML versus JSON WS Interoperability Organization Challenges and Deliverables Profiles Profile Standards Conformance Requirements WS-I Testing Tools Monitor and Analyzer Interoperability Technologies WS-Reliable Messaging WSDL Reliable Messaging Levels of Delivery Assurance Bootstrapping and Configuration Message Transmission Optimization Web Service Security Security at Transport level Security at XML Level XML Encryption XML Digital Signature XML Key Management WS-Security Security Enabled SOAP REST Security Access Restriction Restrict HTTP Methods JSON Web Tokens API Keys

## JAV450: Course Java EE Overview

Code: JAV450

Duration: 1 day

## Audience Course Java EE

The course Java EE Overview is intended for developers, designers, managers and architects who want to get an overview of the capabilities and operation of the Java EE, Enterprise Edition, platform.

## Prerequisites C++, Java or. NET

To participate in this course knowledge of modern software technologies such as C++, Java or. NET and Web applications is desirable.

### **Realization Training Java EE Overview**

The theory is discussed by means of presentation slides. The concepts are illustrated with demos and there is opportunity to practice. The course material is in English.

## **Course Java EE Overview**

The course Java EE Overview discusses the main points of the Java EE standard as it is implemented in application servers like GlassFish, WebSphere and JBoss. The demands of enterprise applications such as scalability, failover and distribution are discussed and how these demands are met by the Java EE platform. Attention is paid to the role of the various Java EE Application Components as Java Servlets, Java Server Pages and Enterprise Java Beans. In this respect the JSF Framework for Java Web Applications is also discussed. Key Java EE services such as JNDI (Java





## Java EE Overview





Naming and Directory Interface) and JTA (Java Transaction API) are part of the subject matter. The various options for accessing databases are treated as well. Attention is paid both to the SQL-based Java Database Connectivity (JDBC) technology and to the new Persistence API for storing objects directly. Furthermore other Java technologies in the context of Java EE, such as Web Services based on SOAP and REST are discussed as well. During the day several application servers and Enterprise Java Bean containers that rely on the Java EE standard are addressed. If time permits, JMX, Java Management Extensions, as an optional module, is treated.

Module 1 : Java EE Intro	Module 2 : Servlets, JSP and JSF	Module 3 : Enterprise Java Beans (EJB)
Java Editions	Servlets and JSP's	Types of Enterprise Beans
Enterprise Application Challenges	Translation and Request Time	Distributed Object Foundation
Java EE Standard	Problems with Servlets and JSP	Architecture of an EJB
Java EE Servers	Classic MVC Pattern	Enterprise Bean Class
Web Components	Model 2 Architecture	EJB Object at work
EJB Components	Using Java Beans	Remote Interface
Java EE and Web Services	Scopes in Web Applications	Session Beans
Deployment Descriptors	ServletContext Scope	Statefull and Stateless
Annotations	Session Scope	Session Bean Pooling
Packaging in EAR Files	Java Web Applications	Message Driven Beans
Java EE Deployment	Web Application Structure	JNDI Naming Context
Configurable Services	MVC Frameworks	Locate resources with JNDI
Java EE API's	Java Server Faces	Context and Dependency Injection
Module 4 : Java EE Persistence	Module 5 : Java EE Web Services	Module 6 : Optional Module : JMX
Java EE Persistence	What is a Web Service?	What is JMX?
Direct File I/O	Web Service Standards	JMX Goal
Serialization	Web Service Types	Where does JMX API fit?
Java Database Connectivity	XML-Schema, SOAP and WSDL	Managed Beans
JDBC Drivers and URL's	JAX-WS Web Services	JMX Architecture
Transparant Persistence	Servlet Based Endpoint	Management Consoles
Object Relational Mapping	Stateless Session Bean Endpoint	Protocol Adapters
Persistence API	JAX-WS Annotations	Standard MBeans
Entity Classes	REST Web Services	Implementing MBeans
Entity Manager	Standard HTTP Methods	Naming MBeans
Persistence Context	Resource URI Access	MBean Server
Entity Managers	Content Negotation	Registering MBeans

## JAV500: Course Advanced Java Programming

Code: JAV500

### Duration: 4 days

#### Price: € 2150

### Audience Course Advanced Java Programming

The course Advanced Java Programming is intended for experienced Java developers who want to gain more in depth knowledge of Java.

### Prerequisites Course Advanced Java Programming

Knowledge of the Java language and syntax and basic experience in **Java** programming is required to participate in this course.

### **Realization Training Course Advanced Java Programming**

The theory is treated on the basis of presentations and is interspersed with exercises. Demos are used to clarify the theory. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Course Advanced Java Programming**

Participants receive an official certificate Advanced Java Programming after successful completion of the course.

## **Course Advanced Java Programming**

In the course Advanced Java Programming a series of advanced aspects of Java are discussed. The course covers the topics that are asked for on the Oracle Certified



## **Advanced Java Programming**





Java Professional or OCP exam and is a good preparation to pass this exam. In the first place attention is paid to aspects of Advanced Class Design such as the implementation of inheritance and composition, the use of polymorphism, interfaces, inner and anonymous classes and the singleton pattern. Next multithreaded applications are discussed and the synchronization between threads when accessing shared data. During the treatment of the concurrency package advanced synchronization mechanisms such as cylic barriers and countdown latches are discussed. Also the functional language constructs introduced in recent Java versions are discussed with lambdas and functional interfaces. Next generics are on the course program with which classes and methods can be parameterised, strong typing is imposed and the chance of runtime errors is limited. Generics are used a lot in the Collection Framework and the most important container classes in this Framework are discussed. Next attention is paid to the <u>Stream API</u> that enables transformations on data collections to be performed by a combination of successive simpler methods like map and reduce. The various possibilities for dealing with errors and exceptions are also on the program and attention is paid to file I/O and new I/O when accessing files and directories. Database access with Java Database Connectivity (JDBC) is treated and queries, prepared statements and transactions are part of that. Finally, if time permits, reflection is optionally on the course program, with which compiled Java classes can be analyzed by means of software, and optionally various aspects of enhancing the Java performance are discussed.

Module 1 : Advanced Class Design	Module 2 : Multiple Threads	Module 3 : Concurrency
Encapsulation and Inheritance Implementing Composition Polymorphism Singleton Patterns Immutable Classes Abstract Classes Final Classes Inner Classes Static Inner Classes Anonymous Inner Classes Autonomous Classes Extending Interfaces Enumerated Types Implementing hashCode and equals	Java Thread Model Extending Thread Class Implementing Runnable Daemon Threads Thread Alive States Thread Class Methods Sleeping and Yielding Control Using join and interrupt Concurrent Method Activation Synchronized Statement Locking and Statics Deadlock Condition Synchronization Using wait and notify	Concurrency Package Task Scheduling Framework Executor Interface ExecutorService Callables and Futures ScheduledExecutorService Synchronizers Semaphores and Exchanger CountdownLatch and CyclicBarrier Concurrent Collections BlockingQueue Interface Lock Interface Reentrant Locks Atomic Variables
Module 4 : Lambda's and Functional Interfaces	Module 5 : Generics and Collections	Module 6 : Streams
Passing Functionality Lambda Expressions Lambda Syntax Lambda Variable Access Lambda Scoping Rules Functional Interfaces Predicate Interface Consumer Interface Supplier Interface Function Interface UnaryOperator Interface BinaryOperator Interface BinaryOperator Interface Method References @FunctionalInterface annotation User Defined Functional Interfaces	What are Generics? Type Erasure and Raw Types Generics and Subtyping Bounded Type Parameters Wildcards Generics in Collections ArrayList and LinkedList TreeSet and Hash Set HashMap and TreeMap ArrayDeque objects Comparable and Comparator Interface Collections Streams and Filters Iteration using forEach Filtering using Lambda's Stream Interface and Stream Pipeline	What are Streams? Lazy Evaluation and Parallelization Core Stream Methods forEach, Map and Filter findFirst and findAny toArray and collect Optional Class Limiting Stream Size allMatch and anyMatch Number Specialized Streams Reduction Operations Parallel and Infinite Streams collect Method Grouping with Collectors class Using flatMap Method
Module 7 : Exception Handling	Module 8 : Java IO and NIO	Module 9 : Database Access met JDBC

Errors and Exceptions	Standard I/O Streams	JDBC Architecture
Exception Handling Syntax	Stream Types	JDBC Drivers and URL's
Checked and Unchecked Exceptions	Reading and Writing Files	Database Connections
Exception Hierarchy	Buffered Streams	Executing Statements
Multiple Catch Clauses	Data Conversion Streams	Querying Databases
finally Clause	Serialization	Update Statements
try with Resources	Object Streams	Retrieving Results
Auto Closeable Resources	NIO and Asynchronous I/O Processing	Handling Errors
Common Exceptions	Working with Buffers	Prepared Statements
Throwing Exceptions	IO Channels	Database Metadata
User Defined Exceptions	Stream API with NIO.2	Transactions
Chained Exceptions and Stack Traces	Using Path Class	Commit and Rollback
Rethrowing Exceptions	Directory Traversing	Rowset Interfaces
Assertions	PathMatcher class	Using RowsetProvider
Module 10 : Localization	Optional Module 11 : Reflection	Optional Module 12 : Performance
LocalDate Class	What is Reflection?	Influences on Performance
LocalTime and LocalDateTime	Reflection Classes	JIT Compilation and Hotspot JVM
Instant and Period	Class Loading	Garbage Collection
Duration and TemporalUnit	The Class Class	String Types
Defining Properties	Creating Objects	Buffered and New I/O
Reading Property Files	Reflection Methods in Class	Synchronization and Concurrency
Creating Resource Bundles	Field Class	Primitives versus Wrappers
Formatting Date and Times	Constructor Class	Collections
Locale Class	Method Class	Exception Handling
Localizing Dates	AccessibleObject Class	Serialization
Localizing Numbers	Dynamic Proxies	Native methods
Localizing Currencies	Invocation Handler	Lazy Loading and Object Reuse

## JAV550: Course Java Performance Improvement

Code: JAV550

Duration: 2 days

## Audience Course Java Performance Improvement

Experienced Java developers who want to improve the performance of Java software.

### **Course Java Performance Improvement Prerequisites**

Knowledge of and experience with Java programming is required to join this course.

## **Realization Training Java Performance Improvement**

The course has a hands-on nature. The theory is treated on the basis of presentation slides and is interspersed with practical exercises. Demos are used to clarify the theory. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Java Performance Improvement**

Participants receive an official certificate Java Performance Improvement after successful completion of the course.

### **Course Java Performance Improvement**

The course Java Performance Improvement will teach you how to analyze and improve the performance of Java applications. In the first place various aspects of

performance in general are treated, such as perceived performance and memory footprint and then aspects of Java performance in particular such as the hotspot JVM and garbage collection. Next attention is paid to the different benchmarks for Java and the various phases of the performance process such as performance monitoring, profiling and tuning. Also a number of specific performance techniques that can be applied to Java code are discussed. The performance impact of various constructs in the Java language and various classes of the Java API are discussed as well. Furthermore, you learn how to deal with performance bottlenecks. In this respect attention is paid to profiling points, time measurements and the detection of memory leaks. Stress testing Web Applications is discussed as well. Finally, special attention goes to performance issues with JDBC and performance in a Java EE environment. Herewith the configuration of pools, caches and the use of load balancing and clustering techniques are treated.

Module 1 : Java Performance Intro	Module 2 : Java API Performance	Module 3 : Performance Profiling
What is Performance? Influences on Performance Important Performance Aspects History of Java Performance JIT Compilation and JIT Compiler Client and Server Hotspot VM Garbage Collection Garbage Collection Algorithms Java Performance Myths Perceived Performance BenchMarking Monitoring and Profiling Performance Tuning Heap Tuning Application Performance Process Heap Activity Monitoring Common Performance Problems Java Performance Tips	Java API Performance Pitfalls String Handling Buffered I/O New I/O Synchronization Concurrency Package Primitives versus Wrappers Collections Array Copying Exception Handling Serialization Native methods Lazy Loading Object Reuse	Profiling Profiling tools Netbeans Profiler Sun Studio Analyzer CPU profiling Approach Profiling a subset Profiling Points Type of Profiling Points Monitoring Threads Lock contention Identifying problematic patterns Stress Testing
Module 4 : Tuning Garbage Collection	Module 5 : JDBC and Java EE Performance	
GC and Performance Java Object Behavior Heap Space Organisation Heap Area Characteristics Young Generation Layout GC Performance Metrics Used GC Algorithms Performance Considerations Serial collector Parallel collector Parallel compact collector Concurrent Mark-Sweep (CMS) Ergonomics	JDBC Optimization Optimization Techniques JDBC connection pooling Single Batch Transactions Smart Queries Tuning Servlets and JSP's HTTP Session Tuning Web Server Tuning Clustering Clustering Types Load Balancing Sticky Sessions	





## Java Performance Improvement







## JAV600: Course Java for Managers

Code: JAV600

Duration: 3 days

## Audience Course Java for Managers

Managers of Java software teams, project leaders of Java software projects, junior Java Developers and other interested persons who want to gain insight in Java software.

## **Prerequisites Course Java for Managers**

General basic knowledge of computer systems and software development is required. Programming experience is an advantage in following this course.

## **Realization Training Java for Managers**

The theory is treated by means of presentation slides and is interspersed with exercises. Demos are used to clarify the theory. The course has a hands-on nature. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Java for Managers**

Participants receive an official certificate Java for Managers after successful completion of the course.

#### **Course Java for Managers**

In this course the basic principles of object oriented programming in the Java

Java Ja tritypeter



## Java for Managers





language and the operation and capabilities of Java technology are discussed. Attention is paid to both client and server technology. The course is always aimed at making clear the principles without going into details. Both the syntax and control flow constructs of Java are discussed as well as object oriented things such as classes, inheritance, interfaces and exceptions. In this respect also UML as modeling language for software systems is on the course schedule. Regarding client technology both Rich Client Applications with GUIs as well as applets and Java Web Start applications are discussed. Regarding server technology the focus is on Java Enterprise applications can store data is on the course schedule. Attention is paid to direct file I/O, serialization, JDBC, Hibernate and the Persistence API. Finally, various Java integration technologies like RMI, JMS and Web Services are discussed as is the Java Micro Edition (Java ME) for mobile phones applications. The course contents covers the requirements of the Java Associate exam 1Z0-850.

Module 1 : Java Platform	Module 2 : Java Language	Module 3 : Object Orientation and UML
History of Java Java Overview Java Editions Java Platform Java Community Process Java Libraries Java Language Compiling Java Programs Running Java Programs Compiler and Interpreter Application structure Garbage Collection Packages Import statement	Java Comments Variables Types of Variables Primitive Data Types Block Statements Operator Precedence Flow Control if else Statements switch Statement for and while Loop do while Loop break and continue Arrays Enhanced For Loop Strings Formatted Output	Programming Paradigms Procedural Programming Object Oriented Programming Object Oriented Paradigm Classes and Objects Example Class and Objects Abstraction and Modularity Encapsulation and Data Hiding Inheritance and Polymorphism What is UML? UML Diagrams Use Cases Sequence Diagrams Class Relationships Generalizations Associations Aggregations Class Diagrams
Module 4 : Java Classes and Interfaces	Module 5 : Rich Client Applications	Module 6 : Java Enterprise Applications
Java Class Definition Access Modifiers Constructors Creating Objects Fields and Methods Overloading and Overriding Using Objects this keyword extends keyword Abstract Classes Interfaces Exceptions Packaging in JAR Threads Java Beans	Java GUI's Abstract Window Toolkit AWT Structure GUI Design with AWT AWT Containers Window Containers AWT Controls Swing AWT versus Swing Swing Component Hierarchy Layout Managers Event Delegation Model Java Applets Applet Parameters Java Plug-In Java Web Start Network Launching Protocol	Java EE Enterprise Challenges Java EE Standard Java EE Servers Web Components EJB Components Persistent Entities Java EE and Web Services Container Services Java EE Deployment Annotations Packaging in EAR Files Development Roles Java EE API's
Module 7 : Servlets and JSP's	Module 8 : Java Persistence	Module 9 : Integration Technologies

- What is a Servlet? Possible Servlet Tasks HTTP Protocol Form Submissions Concurrent Access What is a JSP? JSP Translation Time JSP Request Time Scopes in Web Applications ServletContext Scope Session Scope Classic MVC Pattern Model 2 Architecture Web Application Structure
- Java Persistence Traditional Persistence Transparent Persistence Direct File I/O Serialization Java Database Connectivity JDBC Architecture Problems with JDBC Hibernate Persistent Classes Mapping Files Persistence API Entities Entity Lifecycle
- Remote Method Invocation RMI Architecture Remote Interface and Object Stubs and Skeletons Remote Object Use Java Message Service Messaging Models Point to Point Publish and Subscribe JMS Programming Model Web Services SOAP and WSDL WSDL and Code Generation JAX-WS SAAJ

## Module 10 : Java Mobile Applications

Java ME Back to the Roots Organization of Java ME Configurations and Profiles CLDC Configuration CDC Configuration Profiles Java ME Optional Packages MIDP Profile JWTI Specification Mobile Information Device Profile MIDlets MIDlet Life Cycle States MIDlet Life Cycle Methods MIDlet Build Cycle

## JAV650: Course Java for Application Managers

Code: JAV650

Duration: 2 days

## **Audience Course Java for Application Managers**

The course Java for Application Managers is intended for persons responsible for the monitoring, management and troubleshooting of Java Applications and for other interested persons.

## Prerequisites Course Java for Application Managers

General basic knowledge of computer systems and software development. Programming experience is an advantage in following this course.

### **Realization Training Java for Application Managers**

In this hands-on course the theory is treated by means of presentation slides and is interchanged with exercises. Demos are used to clarify the theory. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Java for Application Managers**

Participants receive an official certificate Java for Application Managers after successful completion of the course.

## **Course Java for Application Managers**

In the course Java for Application Managers the foundations of the Java SE and EE

platform, the Java language and the options to manage and monitor Java software are discussed. The participants become familiar with Java applications and their data types, the object oriented nature of Java, the packaging of Java applications, the principles of garbage collection and the Java thread model. A key module in the course treats the various logging mechanisms in Java software and the configuration of logging. Also attention is paid to principles of exception handling in Java and how stack traces can be interpreted. The Java EE standard is discussed and attention is paid to Java EE Web Components like servlets and JSP's. In this respect the reference implementation for Web Components Tomcat is treated. The participants also become familiar with Java Management Extensions (JMX) as a standard and API for the (remote) management and monitoring of Java Applications. The principles of memory management in Java and the various options to configure garbage collection are discussed as well. The final subject of the course is the performance monitoring and performance tuning of Java applications. Optional appendixes about Java Database Connectivity (JDBC) and the Java Messaging Service (JMS) are provided and will be discussed if this is the desire of the class.

Module 1 : Java Intro	Module 2 : log4j Logging	Module 3 : Stack Tracing
Java Platform	Logging in Java	Error Conditions
Java Editions	log4j characteristics	Exceptions in Java
Java Libraries	log4j Basic Concepts	Exception Handling
Types of Java Applications	java.util Logging	Generated Stack Traces
Compiling and Running Programs	Logging API	Finally Clause
Standalone Application structure	Simple Logging	Exception information
Java Variables	Logging Configuration	Predefined Exceptions
Primitive Data Types	log4j properties	Multiple catch clauses
Classes and Objects	Configuration Options	ArrayIndexOutOfBoundsException
Inheritance	Logger S	NullPointerExceptions
Casting Objects	Logger Output Hierarchy	ClassCastExceptions
Packages	Inheriting Logging Levels	ClassCastExceptions
Packaging in JAR files	Logger Names	Creating Exception Classes
Garbage Collection	Log Levels	Throwing Exceptions
Java Thread Model	Appenders	Chained Exceptions
Thread Characteristics	Layouts	Assertions
Module 4 : Java EE	Module 5 : Java Management Extensions	Module 6 : Memory Management
Java EE Standard	What is JMX?	JVM's Internal Architecture
Java EE Servers	JMX Goal	Heap and Stack
Servlets and JSP's	Where is JMX used	Java Memory Management
Translation and Request Time	Managed Beans	Object Lifecycle
EJB Components	MBean flavors	Strong Object References
Java EE API's	JMX Architecture	Invisible and Unreachable
Apache Tomcat	Java SE Mbeans	Circular References
Tomcat Directories	Naming MBeans	Tuning Garbage Collection
Configuration Files	MBean Server	Generational GC
Web Application Structure	Registering Mbeans	Heap Space Organization
Deployment Descriptor	Manipulating MBeans	Tuning Garbage Collection
Sessions	Notifications	GC Algorithms
Tomcat Logging	Notification Listener	Finalization
Module 7 : Java Performance Tuning	Optional Appendix : JDBC	Optional Appendix : JMS
Influences on Performance	Java Database Connectivity (JDBC)	What is JMS?
JIT Compilation	JDBC Overall Architecture	JMS Terminology
Hotspot JVM	JDBC Operation	JMS Programming Model
Monitoring, Profiling, Tuning	JDBC Drivers	Message Consumption
String Handling	Database URL's	Messaging Domains
Buffered and New I/O	ClassNotFoundException	Queues
Synchronization	Using Tomcat and JDBC	Topics
Collections	Configuring JNDI JDBC Resources	Message Types
Serialization	Context.xml in META-INF	Message Headers
Lazy Loading	JDBC in Web Applications	Durable Subscriptions



Monifo

## Java for Application Managers

## **JAV700: Course Java Data Access Hibernate**

Code: JAV700

Duration: 3 days

## Audience Java Data Access Hibernate Course

Experienced Java developers who want to use Hibernate for accessing data in databases.

### Prerequisites Course Java Data Access Hibernate

Experience with Java programming and object orientation is required. Knowledge of database structures and SQL is beneficial for a proper understanding.

### **Realization Training Java Data Access Hibernate**

The concepts are treated by means of presentation slides and demos. The theory is interspersed with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Java Data Access Hibernate**

Participants receive an official certificate Java Data Access with Hibernate after successful completion of the course.

### **Course Java Data Access Hibernate**

This course addresses Object Relational Mapping with Hibernate. After an overview of the data access capabilities in Java, including JDBC, and the challenge they face,

the basic concepts and architecture of the Hibernate Framework is discussed. The role of the Hibernate configuration file is examined and the mapping of Java classes to database tables is discussed. Next attention is paid to the role and structure of the XML mapping files and the role of the various properties and attributes in these files. The central position of the Hibernate Session created through the SessionFactory is discussed and attention is paid to the various states that Java objects can have in relation to the database like persistent, transient and detached. The various key generation strategies are discussed and also the mapping of association and inheritance relationships to the database are part of the subject matter. Next attention is paid to the capabilities of Hibernate Query language, HQL, to Hibernate criteria and the use of native SQL queries. Finally also Hibernate transactions are discussed, the use of annotations as an alternative to XML mapping files is addressed and the different varieties of Hibernate caching are explained.

Module 1 : Java Persistence	Module 2 : Hibernate Basics	Module 3 : Mapping Persistent Objects
Java Persistence Traditional Persistence Transparent Persistence Persistence Technologies Direct File I/O Serialization Java Database Connectivity JDBC Architecture Executing Statements Retrieving Results JDBC Drivers JDBC URL's Problems with JDBC	What is Hibernate? Hibernate Characteristics Hibernate Configuration Hibernate Configuration File Persistent classes Mapping Files Hibernate Architecture Hibernate Core Concepts Storing Objects Generated Table and SQL Primary Key Column Lifecycle States Persistence Lifecycle	POJO's and JavaBeans equals and hashcode Basic Mappings Class to Table Mappings Property Mapping Identifiers and Generators Multiple Table Mappings Hibernate SessionFactory Hibernate Session Entities and Values Entity Lifecycle Persistent State Transient State Persistent Object Updates Automatic Dirty Checking Detached State Deleting Objects Merging
Module 4 : Mapping Relationships	Module 5 : Mapping Inheritance	Module 6 : Queries with HQL and Criteria
Type of associations Many-to-one Bidirectional Many-to-one Mapping to List, Map Mapping to Bag and Array Using Comparator One-to-one Bidirectional One-to-one Many-to-many Bidirectional Many-to-many Many-to-many Identifier Bag Value Type Collections Collections of Components Sorting Collections Cascading over associations Lazy versus Eager Loading Proxies Detached Objects and Proxies Polymorphic Associations	Inheritance Mapping Strategies Single Table per Class Hierarchy Single Table Data Model Discriminator Columns Advantages and Disadvantages Table per Concrete Class Strategy Table per Concrete Class with unions Table per Class Data Model Advantages and Disadvantages Joined Subclass Strategy Joined Data Model Polymorphism Choosing an Inheritance Strategy	Hibernate Fetching Options Hibernate Query Language HQL Parameters Named Queries Native SQL Criteria Restrictions Query By Example Scrolling and Pagination Query Hints Query Option Pros/Cons N+1 Selects Join Fetching Subselect Fetching Batch Fetching Queries and Fetching Strategies Cartesian Product Problem
Module 7 : Transactions and Concurrency	Module 8 : Hibernate Annotations	Module 9 : Hibernate Configuration









Java Transaction API JTA versus JDBC Transactions Transaction Configuration Hibernate Transaction API Transaction handling pattern Concurrency Isolation Levels Optimistic Locking Versioning Pessimistic Locking ThreadLocal Transactions Conversations Session Lifetime Concurrent Access Metadata Annotations Pros/Cons Configuring Hibernate Annotations Entity and table annotations Primary key annotations Column annotations Special Relation annotations Join column annotations Components Inheritance EJB3/JPA Annotations Connection Pools The promise of Cache Hibernate Caching Architecture First Level Cache Second Level Cache Cache Concurrency Configuring Second Level Cache Cache Regions Eviction

## JAV750: Course Java Data Access with JPA

Code: JAV750

#### Duration: 3 days

## **Audience Course Java Data Access**

Experienced Java developers who want to learn how to use the Java Persistence API for accessing data in databases.

### Prerequisites Java Data Access with Persistence API

Experience with Java programming and object orientation is required to participate in this course. Knowledge of database structures and SQL is beneficial for a good understanding.

### **Realization Training Course Java Data Access**

The concepts are treated on the basis of presentation slides and demos. The theory is interspersed with exercises. All topics in the Java EE 6 Persistence API Developer Certified Expert Exam (1Z0-898) will be discussed. The course material is in English. The course times are from 9.30 up and to 16.30.

### **Certification Java Data Access with Persistence API**

Participants receive an official certificate Java Data Access with Persistence API after successful completion of the course.

### **Course Java Data Access with Persistence API**



## Java Data Access with Persistence API



In this course Object Relational Mapping with the Java Persistence API 2.0 is on the agenda. After an overview of the data access capabilities in Java, including JDBC, and the challenge they face, the basic concepts and architecture of the JPA is explained. The role of the Entity Manager and the persistence.xml configuration file is discussed and attention is paid to the differences between version 1.0 and 2.0 of the JPA specification. The next topic covers the concept of a JPA Entity, the lifecycle of Entities and how Entity classes are mapped to database tables. Also the various states Entities can have in relation to the database such as new, persistent, detached and removed are discussed in this respect as is the concept of merging. Furthermore the use of annotations and XML mapping files and the role the various properties and attributes play is explained. The various key generation strategies are discussed as well as the mapping of association relationships and the mapping of inheritance relationships between Entities. Next the possibilities of the JPA Query language, JPQL, are covered and the uses of JPA criteria and native SQL queries. The function and operation of Entity callbacks that are called immediately before and after the execution of a persistence operation is explained and the alternative use of Entity Listener classes as well. Part of the course program is also the treatment of interceptors which are used for crosscutting concerns like logging and security. Finally, attention is paid to the use of JPA in a Java EE application, how to interact with EJBs and the method of packaging JPA entities. Finally JPA transactions are addressed in both a desktop environment and a Java EE environment.

Module 1 : Intro Java Persistence	Module 2 : Java Persistence API	Module 3 : Mapping Persistent Objects
Java Persistence Traditional Persistence Transparent Persistence Persistence Technologies Direct File I/O Serialization Java Database Connectivity JDBC Architecture Executing Statements Retrieving Results JDBC Drivers JDBC URL's Problems with JDBC	Object Relational Mapping Java Persistence API JPA Versions Entity Classes Entity Manager Persistence Context Entity Identity Entity Lifecycle Entity Relationships Persisting Objects Removing Objects Merging Objects Managing Identity	Mapping Annotations Table Annotation UniqueConstraint Annotation Column Annotation Id Annotation IdClass Annotation GeneratedValue Annotation Version Annotation Basic Annotation Lob Annotation Temporal Annotation Enumerated Annotation Transient Annotation
Module 4 : Mapping Relationships	Module 5 : Mapping Inheritance	Module 6 : JPA Query Language
Entity Relationship types Bidirectional OneToOne Bidirectional ManyToOne Bidirectional OneToMany Bidirectional ManyToMany Unidirectional OneToOne Unidirectional ManyToOne Unidirectional ManyToMany Cascading Persist Cascading Merge Cascading Remove	Mapping Class Hierarchies Mapping Strategies Single Table Inheritance DiscriminatorColumn Settings Single Table per Hierarchy Single Table Mapping Features Joined Subclass Strategy InheritanceType Joined Joined Subclass Per Hierarchy Table per Concrete Class Strategy Abstract Entity Classes Mapped Superclasses	Java Persistence QL JPA QL Characteristics Query Interface Projections Subqueries Joins Update and Delete Queries Dynamic and Static Queries Criteria API Query Error Detection CriteriaBuilder Metamodel in JPA
Module 7 : Entity Callbacks and Listeners	Module 8 : Interceptors	Module 9 : Java EE integration
Life Cycle Callback methods Entity Listeners Life Cycle Callback Rules Signature Life Cycle Callbacks Signature Entity Listeners @PrePersist and @PostPersist @PreRemove and @PostRemove @PreUpdate and @PostLoad Multiple Invocation Callbacks Invocation Order	Interceptor Invocation Model @Interceptor Annotation Interceptor Classes Invocation Context @AroundInvoke Interceptor Lifecycle Interceptor Types Default and Exclude Interceptors PostConstruct or PostActivate PreDestroy and PrePassivate	Enterprise Java Beans Sessions Beans Statefull and Stateless JNDI lookups EJB injection Transaction-Scoped Persistence Context Extended Persistence Context Persistence Unit Packaging in EAR files Deployment Descriptors

## Module 10 : Transactions

Data Integrity Transaction Control Begin, Commit and Rollback Demarcating Boundaries Container Managed Bean Managed Client Managed Transaction Attributes Session Synchronization JTA Transactions Before Completion After Completion

## JAV800: Course Java Development with Spring

Code: JAV800

Duration: 5 days

## Audience Java Development Spring Course

Experienced Java developers who want to use the Spring Framework for Java Desktop, Web and Enterprise applications.

### Prerequisites Course Java Development Spring

Experience with Java programming and object orientation is required to participate in this course. Basic knowledge of Web Applications and XML is beneficial for a proper understanding.

## **Realization Training Java Development Spring**

The concepts are treated by means of presentation slides and demos. The theory is interspersed with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Java Development Spring**

Participants receive an official certificate Java Development with Spring after successful completion of the course.

### **Course Java Development Spring**

This course covers the concepts, components and architecture of the Spring





## Java Development with Spring





Framework. Ample attention is paid to the concept of Dependency Injection or Inversion of Control that plays a central role in Spring. Different varieties of such dependency injection like setter injection and constructor injection are discussed as is the automation of this bean wiring process through autowiring. The other pillar of the Framework, Aspect Orientation, is also addressed. The concepts surrounding Aspect Orientation like Aspects, Joinpoints, Pointcuts, Advice and Weaving are explained. Next the different options to store data from Spring Java applications in databases are treated. Attention is paid to using JDBC with a JdbcTemplate, as well as to the use of Object Relational Mapping frameworks like Hibernate with a Hibernate Template or JPA through annotations. In this respect also Transactions in Spring Rest Web Service are also part of the course schedule and the role of controllers, views, page parameters and command objects is discussed. Finally, attention is paid to Spring Security and the easy setup of a Spring environment with Spring Boot. The modules Spring and JMS and Spring and JMX are optional.

Module 1 : Spring Introduction	Module 2 : Dependency Injection	Module 3 : Application Configuration
What is Spring? Addressing Layers Characteristics Framework Overview Dependency Injection Inversion of Control Aspect Oriented Programming Portable Service Abstractions Spring Packages	Non-loC or Dependency Injection Benefits of Dependency Injection Constructor Dependency Injection Setter Dependency Injection Bean Factory XmlBeanFactory Bean Configuration File Injection Parameter Types Bean Naming Autowiring Properties Application Context Multiple configuration files Working with interceptors Externalizing constant values Bean scopes	Bean definition inheritance Inner beans p and util namespaces Dependency injection of collections Spring Expression Language Autowiring and component scanning Stereotype annotations Java-based configuration Mixing configuration styles When to use XML, annotations, and Java configuration Testing Applications
Module 4 : Aspect Orientation	Module 5 : Spring Persistence	Module 6 : Spring JDBC
Aspect Oriented Programming The need for AOP Crosscutting Concerns Aspect Joinpoints Pointcuts Advise Weaving Target Introduction Spring AOP Static AOP Dynamic AOP Proxies ProxyFactory	Spring and Persistence Java Persistence Traditional Persistence Transparant Persistence Shared Persistence Concepts DAO Design Pattern Before and after DAO DAO Pattern JDBC Integration with IoC DAO Portability Spring DAO Concepts Transaction Management Spring Exceptions Exception Translation	Spring and JDBC JDBC Characteristics JDBC Architecture Executing Statements JDBC Drivers and URL's Spring JDBC Data Access DAO with JdbcTemplate Data Source Injection Querying using JdbcTemplate RowMapper Querying and Populating Objects Updating with JdbcTemplate ResultsetExtractor Callbacks SimpleJdbcTemplate NamedParameterJdbcTemplate JdbcDaoSupport
Module 7 : Spring ORM	Module 8 : Transactions	Module 9 : Spring MVC

Spring and Hibernate Hibernate Integration Mapping Classes Hibernate Template Implementation Hibernate Template Hibernate Template execute Hibernate DAO Implementation Hibernate Annotations Spring and JPA LocalEntityManagerFactoryBean Using JPA API Persistence Unit Configuration LocalContainerEntityManagerFactoryBean Persistence Configuration Persistence Configuration Persistence ExceptionTranslationProcessor Container Managed Transactions Externalizing Database Properties Entity Manager from JNDI	Transaction Managers Declaring Transaction Managers Programmatic Transactions Transaction Callback API @Transactional annotation Declarative Transactions Isolation Levels Read-Only Hint Timeouts Declaring a Transaction Manager Configuring transaction propagation Transactions and integration testing	What Spring MVC? Request life-cycle DispatcherServlet URL Handler mapping Matching URLs Matching Content Types Path Variables Request Parameters Headers and Cookies Injectable Method Parameters Form Submissions Command Objects vs. Entities @RequestBody @ResponseBody Producing Responses ResponseEntity Spring MVC Validation
JpaTemplate and JpaDaoSupport JPA Java Configuration		
Module 10 : Spring REST	Module 11 : Spring and Security	Module 12 : Spring BOOT
REST Web Services @RestController HttpEntity and ResponseEntity Default Content Types Default Status Codes @ResponseStatus and HttpStatus Working with XML Working with JSON Multiple Representations Filtering with @JsonView REST Clients RestTemplate Sending HTTP Requests Translating Entities Reading Responses Error Handlers	Spring Security Model Process Behind Security Interceptors Authentication Manager Configuring authentication Intercepting URLs Security tag library for JSPs Security at the method level Customizing the Security filter chain Access Decision Manager Security Based on Roles Security Based on Identity Run-as Manager Custom Login Pages After Invocation Manager XSD Extensions Using Annotations	Convention over Configuration NO XML Spring Boot CLI Building and Deploying an Application Using Templates Gathering Metrics Using Java With start.spring.io Spring Boot Starters Building as a Runnable JAR Data Access with Spring Data Property Support Securing an Application Authentication and Authorization
Module 13 : Optional : Spring JMS	Module 14 : Optional : Spring JMX	
What is JMS? Messaging Characteristics JMS API Messaging Models JMS Architectural Components JMS Message Interfacs Configuring JMS resources with Spring Using the JmsTemplate Message listener containers	What is JMX? JMX API Managed Beans MBean flavors JMX Architecture Naming MBeans MBean Server Registering Mbeans Manipulating MBeans Export MBeans automatically	

## JAV805: Course Java EE Portlet Programming

Code: JAV805

Duration: 2 days

## Audience Course Java EE Portals and Portlet Programming

This course is intended for experienced Java developers and architects who want to develop Java portals.

## Prerequisites Java EE Portals and Portlet Programming

To join this course knowledge and experience in programming Java and Java Web applications is required.

## **Realization Training Java EE Portals and Portlet Programming**

The course has a hands-on nature. The theory is discussed on the basis of presentation slides and is interspersed with practical exercises. Demos are used to clarify the theory. The course material is in English.

## **Course Java EE Portals and Portlet Programming**

In the course Java EE Portals and Portlet Programming you will learn what portlets are and how they can be included in a portal container. The relationship between servlets and portlets is discussed and the participants learn how portlets can be written, tested and installed in a portal. Particular attention is paid to the standard Java portlets that are written according to the JSR168 standard and JSR268 standard. You will learn more about the different modes portlets can have, about personalizing portlets and about the interaction of the portlets with the portal. Also some more

advanced topics are discussed such as secure portlets, the implementation of caching, session data retention and how you can build portlets that adapt to the specified Locale setting.

Module 1 : Portals and Portlet Intro	Module 2 : Servlets and JSP Review	Module 3 : Portlets Basics
What are Portals? Jetspeed Example Portal What is a Portlet? Portlet Container Example Portals Elements of a Portal Page Portal Applications and WebApps Portlets versus Servlets Differences Portlets and Servlets Portlet File Structure Portlet Portal Interaction Portlet Standards	What is a servlet? HTTP Protocol HTTP Request HTTP Response Form submissions Concurrent Access What is a JSP? JSP Translation Time JSP Request Time Scopes in Web Applications ServletContext Scope Session Scope Request Scope Request Scope Request Dispatching Web Application Structure	Generic Portlet Base Class Generic Portlet Subclasses Portlet Interface PortletConfig Interface Portlet Lifecycle Simple Example Portlet Deployment Descriptor Portlet Modes Portlet Window States Portlet Request Processing Action Request and Response Render Request and Response Lifcycle Implications Portlet URL's
Module 4 : JSR168 Portlets	Module 5 : WSRP Portlets	Module 6 : Ajax Review
JSR 168 Portlet Standard JSR 168 Subjects Custom Portlet Modes Custom Portlet Window States PortletContext interface PortletContext interface Portlet Data Portlet Sessions Portlet Preferences Render Parameters User Attributes Example User Attributes PortletRequesDispatcher Portlet Tag Library	What is WSRP? What is a Portlet WSRP Consumer and Producer WSRP Benefits WSRP Interfaces and Operations Service Description Interface Markup Interface Registration Interface Portlet Management Interface Portlet Life Cycle Cloning and Setting Properties WSRP Support Example Application	Classic Web Application Model Ajax Web Application Model Classic Synchronous App. Model Ajax Asynchronous App. Model How Ajax Works Creating the XMLHttpRequest Object Methods Sending the Request Object Properties XMLHttpRequest readyState Listening for Response Processing the Response Sequence diagram
Module 7 : JSR286 Portlets		

Events Shared Render Parameters Resource Serving Ajax Portlet Filters Caching changes Window ID Namespacing Request Dispatcher availability Portlet Taglib additions Additional CSS classes Coordination Cookies and Headers





Price: € 1199

## Java EE Portals and Portlet Programming





## JAV808: Course Java EE Technologies

Code: JAV808

### Duration: 5 days

## Audience Course Java EE Technologies

The course Java EE technologies is intended for System architects and developers who want make the correct Java technology choices for their system and environment and application administrators who want to get a better understanding of various Java technologies.

## **Prerequisites Java EE Technologies**

General basic knowledge of software architecture and knowledge of the Java platform is required to participate in this course.

## **Realization Training Java EE Technologies**

This course has a very practical nature but it is not a programming course. Central is the understanding of the operation of a technology. The theory is interspersed with short case studies. Modern IDE's such as Eclipse and NetBeans and Application Servers like JBoss and Tomcat are used. The course material is in English.

## **Course Java EE Technologies**

The course Java EE Technologies is intended for system architects who are involved in the deployment of Java EE technology. The participants learn how to make the right choices between competing options. The course provides an overview and

comparison of different modern Java EE technologies and brings the knowledge of the participants on a higher level. After an overview of the Java EE platform, the most important components, Servlets and JSPs, are treated. The JSF Framework as MVC lavar on the basis of Sondots and JSPs is also discussed. Furthermore, Single Page Applications are explained and the Aiax technology which is based

as MVC layer on the basis of Servlets and JSPs is also discussed. Furthermore, Single Page Applications are explained and the Ajax technology which is based on asynchronous requests to the server is treated. The essence of various JavaScript frameworks such as Angular, React and Vue are also on the course program. Then we go into Java Management Extensions, JMX, which enable the monitoring and management of Java Applications and Servers. The other important component of the Java EE platform, Enterprise Beans or EJBs, will be discussed as well. And also attention is paid to the options for persistence in Java (JDBC and Persistence API). The modules SOAP and REST Services provide an overview of the Java Web Service technology. The implementation of a Micro Service Architecture and the various frameworks that can be used for this purpose are on the course program as well. The course ends with a discussion of the Java Messaging Service (JMS) and security in the Java EE platform with policies, certificates, authentication, authorization, JSON Tokens, API keys and JAAS.

Module 1 : Java EE Architecture	Module 2 : Servlets and JSP's	Module 3 : Java Server Faces
Java EE Standard Java EE Servers Web Components EJB Components Persistent Entities Standard Java Beans Layered Architecture Container Services Java EE Web Services Deployment Descriptors Annotations Packaging in EAR Files	What is a Servlet? Servlet Initialization HTTP Protocol Form Submission Concurrent Access What is a JSP? Translation and Request Time Scopes in Web Applications ServletContext Scope Session and Request Scope Web Application Structure Classic MVC Pattern	JSF Feature Overview Request Processing Phases Server Side UI Components JSF Component Libraries Deployment Descriptor Faces Configuration File Facelets Page Structure Managed Beans Expression Language Facelet Default Navigation Event Handling Validators and Convertors
Module 4 : Single Page Applications	Module 5 : JMX	Module 6 : Enterprise Java Beans
Classic Web Application Model Ajax Web Application Model Single Page Applications Typical Ajax Interactions Creating XMLHttpRequest XMLHttpRequest Methods XMLHttpRequest Object Properties Sending the Request XMLHttpRequest readyState responseText and responseXML JavaScript Frameworks Angular, React and Vue	Java Management Extensions JMX Goal Where is JMX used Managed Beans MBean flavors JMX Architecture Java SE Mbeans Naming MBeans MBean Server Registering Mbeans Manipulating MBeans Notification Listener	EJB Features Session Beans Statefull and Stateless Architecture of an EJB Remote versus Local Clients Web Service Clients EJB 3.x Programming Model Life Cycle Session Beans Session Bean Pools Activation and Passivation Message Driven beans Life Cycle MDB Beans
Module 7 : Persistence Technologies	Module 8 : SOAP Services	Module 9 : REST Services
Direct File I/O and Serialization JDBC Overall Architecture JDBC Drivers and URL's Object Relational Mapping Persistence API in EJB 3.x Entity Classes Entity Manager Persistence Context Persistence Unit Entity Lifecycle Merging Objects Managing Identity	What is a Web Service? RPC versus Document Style XML-Schema Java XML Mapping Java API XML Binding JAXB Binding Life Cycle JAXB API SOAP Messages Web Service Description Language JAX-WS Service Side Programming Model Client Side Programming Model	What is REST? Standard HTTP Methods ID and Links Reference Implementation JAX-RS Addressing Path Parameters Content Negotation Multiple Representations Stateless Communications Container Item Pattern Map, Key, Value Pattern
Module 10 : Micro Services	Module 11 : Java Messaging Service	Module 12 : Java Security



and

Architecture

What are Microservices? Creating HTTP MicroServices Consuming HTTP MicroServices MicroService Frameworks Spring BOOT Automatic Configuration Application Packaging DropWizard MicroServices Reactive Microservices From Callbacks to Observables Message Based Microservices What is JMS? Messaging Characteristics JMS API Publish and Subscribe Point tot Point JMS Architectural Components Message Types Creating and Receiving Messages Message Driven Beans Message Listeners onMessage method Authentication and Authorization JAAS Declarative Security Programmatic security Form Based Authentication Basic and Digest Authentication Secure Sockets Layer Encryption Types REST Service Security JSON Web Tokens API Keys

## JAV825: Course Wicket Programming

Code: JAV825

#### Duration: 3 days

## Audience Wicket Programming Course

The course Wicket Programming is aimed at companies and individuals who are using or planning to use Wicket and require a detailed understanding of the relevant technologies.

## **Prerequisites Course Wicket Programming**

Professional programming experience in Java and knowledge of Java Web applications is required.

## **Realization Training Wicket Programming**

The subject matter is treated on the basis of presentation slides. Demos are used to clarify the theory. Ample exercises are done during the course. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Wicket Programming**

Participants receive an official certificate Wicket Programming after successful completion of the course.

## **Course Wicket Programming**

The course Wicket Programming focuses on the development of applications using

the Wicket framework. Topics include a Wicket Framework overview, using Wicket components, creating model classes, user input, data conversion and validation, building reuseable web components and AJAX. The course is intended for experienced Java programmers who need to understand and use the Wicket architecture. Participants who complete the Wicket Development course will be able to: Understand the Wicket framework, develop models, validators, and components, develop Web applications using Wicket.

Module 1 : Wicket Intro	Module 2 : Core Concepts	Module 3 : Components
What is Wicket? Component Orientation Wicket Features More Wicket Features Wicket Timeline Component Hierarchy Wicket in Architecture Wicket Configuration Hello Wicket Application Hello Wicket HTML Page General Application Structure Wicket Run Modes Wicket Distribution and Modules Wicket Resources	Wicket Application Request and Response Classes RequestCycle Request Processing RequestCycle Hooks Stateful and Stateless Pages Session Class Components Wicket Models Models and POJO's Link Component Wicket Component Examples Page Rendering Process Behaviors	Wicket Components Component Hierarchy Component Class MarkupContainer class Component Lifecycle Component Hook Methods Method OnBeforeRender Panel Component Border Component HTML Tags and Attributes Repeating Views List Views Behaviors
Module 4 : Models	Module 5 : Forms	Module 6 : Ajax and Rich Components
What are Models? Set and Get Model Using Models Model Factory Methods Models and JavaBeans Using PropertyModel CompoundPropertyModel Using CompoundPropertyModel Wicket Forms Forms and Models Sample Logging Form CompoundPropertyForm Login DropDownChoice Model Model Chaining Detachable Models LoadableDetachableModel	Wicket Forms Form Validation Displaying Feedback Built-in Validators Feedback Messages Custom Validators Converters Input Conversion Validation with JSR303 IFormSubmittingComponent Components Button Nested Forms Complex Forms Stateless Forms Checkboxes ListMultipleChoices	Ajax Explained Classic Web Application Model Ajax Asynchronous Model Wicket and Ajax AjaxLink Setting MarkupID Built-In Ajax Components AjaxEditableLable AutoCompleteTextField Modal Window Ajax Behaviors Using Ajax Behaviors Ajax Request Attributes IAjaxCallListener Ajax Call Listeners Global Listeners
Module 7 : jQuery Integration	Module 8 : Trees and Wizards	Module 9 : New Features
Using jQuery jQuery Function Object jQuery Selection Replacing Elements JQWicket jQuery Library Wicket jQuery UI Calendar Example Features to Implement jQuery UI JQDatePicker.js Initialization Code Header Contribution Code	Trees Tree Components Build Nodes Instantiate Tree Checked Nodes Autocheck Folder CheckModel OnUpdate What is a Wizard Wizard Types Two Panel Wizard UpdatePanel	Wicket 6 Migration Repeaters and Data Form Processing Wicket 7 Migration Feedback Storage Refactoring Header Rendering Strategy IHeaderResponse Changes Wicket 8 Migration Decorator to Listener Wicket Decorator to Listener List of Renamed Classes New Modules





**APACHEWICKET** 

## **Wicket Programming**

## JAV850: Course Advanced Spring Development

Code: JAV850

## Duration: 3 days

## Audience Course Advanced Spring Development

The Advanced Spring Development course is intended for experienced Spring Developers who want to learn advanced aspects of Spring.

## Prerequisites Course Advanced Spring Development

Knowledge of the basic concepts of Spring such as dependency injection and experience with programming in the Spring Framework is required to participate in this course.

## **Realization Training Advanced Spring Development**

The concepts are treated on the basis of presentations and demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification Advanced Spring Development**

Participants receive an official certificate Advanced Spring Development after successful completion of the course.

## **Course Advanced Spring Development**

In the course Advanced Spring Development a number of advanced applications and projects of the Spring Framework are discussed.





## Advanced Spring Development





Module 1 : Spring Internals	Module 2 : Spring Annotations	Module 3 : Spring Boot
What is Reflection? Reflection Classes Class Loading Creating Objects Reflection Methods in Class Field Class Field Class Usage Constructor Class Method Class AccessibleObject Class Dynamic Proxies Invocation Handler	What are annotations? Pre-Java 5 annotations Why annotations? Defining annotation types Using annotations Kinds of annotations Marker annotations Single value annotations Normal annotations Meta-Annotations Retention meta-annotation Retention Class and Runtime	What is Spring Boot? Advantages Spring Boot Goal of Spring Boot Spring Boot Flavors Key Spring Boot Components Spring Boot Starter Starter Dependencies Spring Boot Autoconfigurator @SpringBootApplication Spring Boot CLI Spring Boot Internals Spring Boot Actuator
Module 4 : Spring Data	Module 5 : Spring REST	Module 6 : Spring Batch
What is Spring Data? Spring Data Configuration CRUD Out of the Box JPA Repositories Persisting and Modifying Entities Spring Data Queries @Query Annotation Named and Async Queries Paging Results Customizing Repository Behavior Spring Data MVC Integration MongoDB Template Mapping and Inserting Documents	What is REST? Request Processing Spring MVC @Controller @RequestMapping Methods @ModelAttribute REST Web Services Simple REST Examples REST Web Service Principles @RestController Embedded Path Parameters Json View Resolver Multiple Representations Content Negotation	What is Spring Batch? Concepts and terminology Steps and Items Tasklets and Chunks Readers and Writers Processors and Executions Configuring Spring Batch Creating jobs and steps Creating a JobRepository Scopes and Listeners Scheduling and Shutdown Controlling Execution Scalability and concurrency
Module 7 : Spring JMS	Module 8 : Spring Kafka	Module 9 : Spring Cloud
What is JMS? Messaging Characteristics and Models JMS Architectural Components Spring JMS JMS with Spring Spring JMS Options Spring JMS Options Send and Convert Execute Method ProducerCalback Receive and Convert Message Listener Containers SessionAwareMessageListener MessageListenerAdapter	What is Spring Kafka? Sending Messages KafkaTemplate Transactions ReplyingKafkaTemplate Receiving Messages Message Listeners Message Listeners Containers @KafkaListerner Annotation Container Thread Naming Lifecycle Management Retrying Deliveries Stateful Retry Kafka Streams Support	What is Spring Cloud? Spring Cloud Config Eureka Service Spring Cloud Bus Spring Cloud Cluster Spring Cloud Security Spring Cloud Data Flow Cloud Task Spring Cloud Connectors Spring Cloud Connectors Spring Cloud Task App Starters Spring Cloud Zookeeper Spring Cloud CLI Spring Cloud Gateway Spring Cloud Pipelines

Price: € 1750

## JAV888: Course Spring Boot Development

Code: JAV888

### Duration: 4 days

Price: € 2250

## **Audience Spring Boot Development**

The course Spring Boot Development is intended for experienced Java Developers who want to use Spring Boot for application development.

## **Prerequisites Course Spring Boot Development**

Experience with programming in Java and object orientation is required to participate in this course. Basic knowledge of the Spring Framework is beneficial to good understanding.

## **Realization Training Spring Boot Development**

The concepts are treated on the basis of presentations and demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification Spring Boot Development**

Participants receive an official certificate Spring Boot Development after successful completion of the course.

## **Course Spring Boot Development**

In the course Spring Boot Development participants learn to use Spring Boot in the development of Java applications.





## **Spring Boot Development**



Module 1 : Spring Core	Module 2 : Groovy	Module 3 : Spring Boot
Spring Framework Overview Spring Configuration Spring Dependency Injection Non IoC versus IoC BeanFactory Application Context Beans Life Cycle XML Configuration Configuration with Annotations Component Scanning Spring Java Configuration Aware Interfaces	What is Groovy? Groovy Features HelloWorld in Groovy Removing Noise Removing Boilerplate Dynamic Types Variable Interpolation POJOs on Steroids Script Support Differences with Java Running Groovy Scripts Groovy Development Kit	What is Spring Boot? Advantages Spring Boot Goal of Spring Boot Spring Boot Flavors Key Spring Boot Components Spring Boot Starter Starter Dependencies Spring Boot Autoconfigurator @SpringBootApplication Spring Boot CLI Spring Boot Internals Spring Boot Actuator
Module 4 : Dependency Injection	Module 5 : Application Configuration	Module 6 : Aspect Orientation
Non-IoC or Dependency Injection Benefits of Dependency Injection Constructor Dependency Injection Setter Dependency Injection Autowiring with @Autowired @Qualifier Annotation @Component Annotation @Repository and @Service Bean scopes Event Handling Internationalization	Configuration Classes @Configuration Annotation @Bean Annotation @Enable Annotations @EnableAutoConfiguration Autowiring and Component Scanning @EnableScheduling Wire External Values Spring Expression Language @Value Annotation @PropertySource Annotation	What is AOP? The need for AOP Crosscutting Concerns Traditional Approach Spring AOP AOP Concepts AOP Key Terms Aspects and Weaving Pointcuts and Joinpoints ProxyFactoryBean Spring AOP Configuration
Module 7 : Spring JDBC	Module 8 : Spring Data	Module 9 : Spring REST
Spring and JDBC JDBC Architecture Executing Statements JDBC Drivers and URL's Spring JDBC Data Access Spring DAO with JdbcTemplate Data Source Injection Querying using JdbcTemplate RowMapper Querying and Populating Objects Updating with JdbcTemplate ResultsetExtractor Callbacks NamedParameterJdbcTemplate	What is Spring Data? Spring Data Configuration CRUD Out of the Box JPA Repositories Persisting and Modifying Entities Spring Data Queries @Query Annotation Named and Async Queries Paging Results Spring Data MVC Integration Transaction Handling @Transactional Annotation MongoDB Template Mapping and Inserting Documents	REST Web Services @RestController HttpEntity and ResponseEntity Default Content Types Default Status Codes @ResponseStatus and HttpStatus Working with XML and JSON Multiple Representations Filtering with @JsonView REST Clients RestTemplate Sending HTTP Requests Reading Responses Error Handlers
Module 10 : Spring Security	Module 11 : Spring Cloud	

## JAV900: Course Java FX Programming

Code: JAV900

Duration: 2 days

## Audience Course Java FX Programming

Java developers who need to learn Java FX for designing desktop and mobile front ends.

## **Prerequisites Course Java FX**

To join this course knowledge of Java development is required. Experience with GUI design is beneficial for a proper understanding.

## **Realization Training Java FX Programming**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Demos are used to clarify the theory. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Java FX Programming**

Participants receive an official certificate Java FX Programming after successful completion of the course.

### **Course Java FX Programming**

JavaFX is pitched squarely at the Adobe Flash / Microsoft Silverlight arena of bringing rich interfaces to internet and mobile applications. The course is aimed at Java

developers who need to create front ends to replace AWT and Swing (for desktop) or Java ME and JWT (for mobile). In this course students will learn to create user interfaces using a declarative style enabled by the JavaFX builder classes. Attention is paid to patterns for JavaFX developers and to property binding to keep the UI synchronised with the model. Students will also learn about JavaFX UI controls, charts, shapes, effects, transformations and animations to create stunning, responsive, user interfaces. Also discussed are the JavaFX layout classes to define the user interface in a cross-platform fashion and the observable collection classes to observe changes in, and bind to, Java Collections. Finally JavaFX media classes to play audio and video are treated and the interaction with external application services to create an enterprise application with JavaFX as well.

Module 1 : JFX Intro	Module 2 : Creating a GUI	Module 3 : Properties and Bindings
Introducing JavaFX Minimum effort and maximum impact Comparing Java and JavaFX Comparing JavaFX with Adobe AIR GWT, and Silverlight Deployment and More Packaging and Deployment JavaFX in Swing Interoperability with SWT Use a Doclet	Startup Window Main Window Menu Bar Selection and Message Bar Content Panel Library Panel Hierarchy Panel Inspector Panel Style Sheet Support Internationalization Support	Understanding Properties Defining a Property Using a ChangeListener High-Level Binding API Using the Bindings Class Combining Both Approaches Observable, ObservableValue InvalidationListener ChangeListener Low-Level Binding API
Module 4 : Java FX	Module 5 : JavaFX UI Controls	Module 6 : Collections and Concurrency
Application Logic Architecture and Framework Work with the Scene Graph Use Properties and Binding Work with Collections Concurrency and Threads Implement Best Practices	Work with Layouts Add Text Work with UI Controls Build UI with FXML Handle Events Create Charts Add HTML Content Skin Applications with CSS Drag and Drop Data Work with Canvas Use Image Ops Scene Builder Scene Builder Overview Get Started with Scene Builder Scene Builder User Guide	Java Collections Basics Using a List Using a Map Collections Class JavaFX Collections Using an ObservableList Using ObservableMap FXCollections Change Notifications
Module 7 : Creating Charts in JavaFX	Module 8 : Using the Media Classes	
Introduction to JavaFX Charts Pie Chart Line Chart Area Chart Bubble Chart Scatter Chart Bar Chart Styling Charts with CSS	Effects, Animation, and Media Create Visual Effects Add 2D and 3D Transformations Add Transitions & Animation Incorporate Media	





## Java FX Programming





## JAV906: Course Web Development with GWT

Code: JAV906

Duration: 3 days

## Audience Course Web Development with GWT

The course Web Development with GWT is intended for Java developers who want to create dynamic Web Applications with the <u>Google Web Toolkit (GWT)</u> and who want to explore the potential of this framework.

## Prerequisites Course Web Development with GWT

Knowledge of and experience with <u>Java</u> development and the development of Web applications is required to participate in this course.

## **Realization Training Web Development with Google Web Toolkit**

The theory is discussed on the basis of presentations. Demos are used to clarify the theory. There is ample opportunity to do exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

## **Certification Course Web Development with GWT**

Participants receive an official certificate Web Development with GWT after successful completion of the course.

## **Course Web Development with GWT**

In the course Web Development with GWT participants get an overview of the

operation and possibilities of the Google Web Toolkit (GWT) and learn to use the GWT to create dynamic web applications. The Google Web Toolkit is a framework that enables web developers to write complex JavaScript front-end applications in Java. The GWT compiler takes care of the translation of the Java code into JavaScript. The GWT emphasizes standard procedures for typical web development tasks such as asynchronous remote procedure calls, history management, bookmarking, user interface abstraction, internationalization and cross-browser portability. The course discusses the Google Web Toolkit architecture, the distinction between Development mode and Production mode, the JRE emulation library and deployment of GWT applications. Extensive attention is paid to the GWT GUI class library with various GWT widgets, the event listener model and the use of CSS. The layout of the User Interface through the various GWT panels and declarative in XML is on the course program. Also attention is paid to how in Google Web Toolkit single page functionality is implemented by means of Remote Procedure Calls (RPCs) to server objects. Finally the Java Script Native Interface (JSNI), the communication between Java and JavaScript using JSON and internationalizations are on the program.

Module 1 : GWT Intro	Module 2 : GWT Architecture	Module 3 : GWT Event Handling
What is GWT? Single Page Application Model Synchronous versus Asynchronous GWT Solution Installing GWT Eclipse Plugin Browser Plugin Development Mode Production Mode Deployed mode Creating Projects Deploying Applications	GWT Application Components Main GWT Project Configuration Entry Point Class Module Configuration File HTML Host Page Project Configuration with RPC Compiler Output GWT Production Mode Running on External Server Java to Javascript compiler JRE Emulation library Deferred Binding	Basic Widgets Common Events Pushbuttons Checkboxes and Radiobuttons Listboxes Textfields Implementing Event Handling Event Handler Interface Event Handler Classes Named Inner Classes Anonymous Inner Classes Lambda Functions
Module 4 : Layout with Panels	Module 5 : Extended GUI Widgets	Module 6 : GWT RPC
GWT UI Class Library GWT Widgets and Panels Root Panel Panels for Layout HorizontalPanel and VerticalPanel FlowPanel and Grid FlexTable Composite Panels TabLayoutPanel StackLayoutPanel DockLayoutPanel DeckLayoutPanel	Popup Messages DialogBox PopupPanel Native Dialog Boxes DatePicker Widget Date Input Boxes MenuBar Widget Fancy Text Boxes Autocompleter Text Boxes Optional Text Box Tree Widget TabBar Widget Custom Widgets	What is GWT RPC? Overall RPC Idea RPC Data Types GWT Serialization Communication with Server GWT RPC Mechanism GWT Plumbing Diagram Service Interface Callback Version Service Interface Service Servlet Service Implementation Create Service Proxy Client Side Callbacks
Module 7 : Declarative Layout	Module 8 : JavaScript Native Interface	Module 9 : GWT Internationalization
Java Based Layout Problems UiBinder Approach HTML-based UiBinder GUIs Widget-based UiBinder GUIs UI Event Handling with UiBinder UiBinder XML syntax @UiConstructor @UiField CSS in UiBinder Images in UiBinder	JavaScript Native Interface (JSNI) Where is JSNI used? Format of Methods \$wnd and \$doc variables Basic JSNI Syntax JSNI Arguments and Return Types Passing Values JSNI Fields and Methods JavaScript Object Notation (JSON) JavaScriptObject Overlay Types	Localization of Applications GWT Property Files Deferred Binding Seperate JS files String Internationalization Local Model GWT I18N library DateTime Formatting Number Formatting Setting History Tokens Responding to History Tokens









## JAV950: Course Test Driven Development with JUnit

Code: JAV950

Duration: 2 days

## Audience Test Driven Development with JUnit Course

This course is intended for experienced Java developers who want to apply JUnit for Test Driven Development.

## Prerequisites Course Test Driven Development with JUnit

Knowledge of and experience with programming in Java is required to join this course.

## **Realization Training Test Driven Development with JUnit**

The theory is covered on the basis of presentation slides and is interspersed practical exercises. Demos are used to clarify the discussed concepts. The course material is in Enalish.

## **Course Test Driven Development with JUnit**

The course Test Driven Development with JUnit will teach participants the principles of and reasoning behind Test Driven Development and the role of unit testing therein. After an overview of the different types of testing and their use, detailed attention is given to the workings of the JUnit library, the integration of this library in Eclipse and the use of assertions in Test Cases and Test Suites. Fixtures used for the formulation of pre- and postconditions of Test Cases are also discussed as is the automation of tests and the concept of continuous integration. Next the methodology of Test Driven

Development (TDD) is discussed, the three rules of TDD and the steps in TDD are explained as are the benefits and limitations of TDD. The participants will exercise TDD by trying to solve so called code Kata's, small programming problems, using TDD. After an overview of the importance of writing clean code, the use of stubs and mocks is treated. These stubs and mocks are used as replacement for code that is not ready yet in a testing environment and can be replaced by real code in a production environment. In this respect the Mockito library is used as an example of a mocking framework. Finally attention is paid to database unit testing using DBUnit and the testing of Web Applications using HTMLUnit.

Module 1 : Unit Testing	Module 2 : JUnit	Module 3 : Test Driven Development
What is Unit Testing? Benefits of Unit Testing Manual Testing Automated Testing Time to Test Unit Test Example Unit Testing Best Practises Using Seams Testing Frameworks Other Types of Testing Continuous Integration Regression Testing Usability Testing Exploratory Testing Acceptance Tests Concurrency Bug Concurrency Testing	What is JUnit? JUnit Features JUnit Integration JUnit View in Eclipse JUnit Classes JUnit Classes JUnit Class Diagram Test Cases TestCase Class TestResult Class JUnitCore Assert Statements Assert Class Fixtures Test Suites Annotations Special Cases Testing for Exceptions	What is Test Driven Development? Traditional Testing versus TDD Three Rules of TDD Steps in TDD Test Cycles Benefits of TDD Limitations of TDD Testing versus Design TDD Adaptation Behavior Driven Development Designing for Testing Code Kata's Example Kata Domain Model Kata Test and Implementation
Module 4 : Clean Code	Module 5 : Stubs and Mocks	Module 6 : Database Unit Testing
What is Clean Code? Clean Code Principles Technical Debt Meaningful Naming Naming Guidelines What to Avoid Functions Abstraction Level Switch Statements Function Arguments Avoid Side Effects Command Query Separation Comments Expressing in Code Good Comments Bad Comments Code Smells	Using Test Doubles What are Stubs? Stub Usage Method under Test Stub HTTP Connection Stubbing Web Server Use Embedded Web Server Stubbing Server Resources Using Jetty Handlers Mock Object Simple Mock Example Collaborating Objects Mock Implementation Test using Mock Anti Patterns Using Mockito	Unit Testing Data Access Types of DB Unit Testing Database Integration Unit Testing DB Unit Advantages of DBUnit DB Unit Life Cycle Core Components IDataSet Implementations Concrete Database Operations Presetting Database Data Extract Data From Database DBUnit Fixture Setup Tables and Dataset Data Access Test Cases Abstract Class Fixture

## Module 7 : Web Application Testing

Testing Web Applications What is HTMLUnit HTMLUnit Features Simple HTMLUnit Test Imitating Browsers HTML Form Test Finding Specific Elements Button Click Test



ALL CODE IS GUILTY **UNTIL PROVEN INNOCENT** 

## Java Test Driven Development with jUnit



## **ALL CODE IS GUILTY UNTIL PROVEN INNOCENT**

# JAV960: Course Microservices with Spring Boot

Code: JAV960

## Duration: 3 days

Module 1 : Spring Introduction	Module 2 : Dependency Injection	Module 3 : Application Configuration
What is Spring? Addressing Layers Characteristics Framework Overview Dependency Injection Inversion of Control Aspect Oriented Programming Portable Service Abstractions Spring Packages	Non-loC or Dependency Injection Benefits of Dependency Injection Constructor Dependency Injection Setter Dependency Injection Bean Factory XmlBeanFactory Bean Configuration File Injection Parameter Types Bean Naming Autowiring Properties Application Context Multiple configuration files Working with interceptors Externalizing constant values Bean scopes	Bean definition inheritance Inner beans p and util namespaces Dependency injection of collections Spring Expression Language Autowiring and component scanning Stereotype annotations Java-based configuration Mixing configuration styles When to use XML, annotations, and Java configuration Testing Applications
Module 4 : Aspect Orientation	Module 5 : Spring Persistence	Module 6 : Spring JDBC
Aspect Oriented Programming The need for AOP Crosscutting Concerns Aspect Joinpoints Pointcuts Advise Weaving Target Introduction Spring AOP Static AOP Dynamic AOP Proxies ProxyFactory	Spring and Persistence Java Persistence Traditional Persistence Transparant Persistence Shared Persistence Concepts DAO Design Pattern Before and after DAO DAO Pattern JDBC Integration with IoC DAO Portability Spring DAO Concepts Transaction Management Spring Exceptions Exception Translation	Spring and JDBC JDBC Characteristics JDBC Architecture Executing Statements JDBC Drivers and URL's Spring JDBC Data Access DAO with JdbcTemplate Data Source Injection Querying using JdbcTemplate RowMapper Querying and Populating Objects Updating with JdbcTemplate ResultsetExtractor Callbacks SimpleJdbcTemplate NamedParameterJdbcTemplate JdbcDaoSupport
Module 7 : Spring ORM	Module 8 : Transactions	Module 9 : Spring MVC
Spring and Hibernate Hibernate Integration Mapping Classes HibernateTemplate Implementation HibernateTemplate HibernateTemplate execute Hibernate DAO Implementation Hibernate Annotations Spring and JPA LocalEntityManagerFactoryBean Using JPA API Persistence Unit Configuration LocalContainerEntityManagerFactoryBean Persistence Configuration PersistenceExceptionTranslationProcessor Container Managed Transactions Externalizing Database Properties Entity Manager from JNDI JpaTemplate and JpaDaoSupport JPA Java Configuration	Transaction Managers Declaring Transaction Managers Programmatic Transactions Transaction Callback API @Transactional annotation Declarative Transactions Isolation Levels Read-Only Hint Timeouts Declaring a Transaction Manager Configuring transaction propagation Transactions and integration testing	What Spring MVC? Request life-cycle DispatcherServlet URL Handler mapping Matching URLs Matching Content Types Path Variables Request Parameters Headers and Cookies Injectable Method Parameters Form Submissions Command Objects vs. Entities @RequestBody @ResponseBody Producing Responses ResponseEntity Spring MVC Validation
Module 10 : Spring REST	Module 11 : Spring and Security	Module 12 : Spring BOOT
REST Web Services @RestController HttpEntity and ResponseEntity Default Content Types Default Status Codes @ResponseStatus and HttpStatus Working with XML Working with JSON Multiple Representations Filtering with @JsonView REST Clients RestTemplate Sending HTTP Requests Translating Entities Reading Responses Error Handlers	Spring Security Model Process Behind Security Interceptors Authentication Manager Configuring authentication Intercepting URLs Security tag library for JSPs Security at the method level Customizing the Security filter chain Access Decision Manager Security Based on Roles Security Based on Identity Run-as Manager Custom Login Pages After Invocation Manager XSD Extensions Using Annotations	Convention over Configuration NO XML Spring Boot CLI Building and Deploying an Application Using Templates Gathering Metrics Using Java With start.spring.io Spring Boot Starters Building as a Runnable JAR Data Access with Spring Data Property Support Securing an Application Authentication and Authorization

Module 13 : Optional : Spring JMS	Module 14 : Optional : Spring JMX
What is JMS? Messaging Characteristics JMS API Messaging Models JMS Architectural Components JMS Message Interfacs Configuring JMS resources with Spring Using the JmsTemplate Message listener containers	What is JMX? JMX API Managed Beans MBean flavors JMX Architecture Naming MBeans MBean Server Registering Mbeans Manipulating MBeans Export MBeans automatically

## JAV999: Course Spring 5 Development

Code: JAV999

Duration: 3 days

## Audience Course Spring 5 Development

The course Spring 5 Development is intended for developers with knowledge of and experience with the Spring Framework who want to learn the new features of Spring 5.

## **Prerequisites Course Spring 5 Development**

Experience with programming in an older version of the Spring Framework is required to take part in this course.

## **Realization Training Spring 5 Development**

The concepts are treated on the basis of presentations and demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification Course Spring 5 Development**

Participants receive an official certificate Spring 5 Development after successful completion of the course.

## **Course Spring 5 Development**

In the course Spring 5 Development the new features of the Spring 5 Framework are discussed.





## Spring 5 Development





Module 1 : Spring 5 New Features	Module 2 : Spring Core Additions	Module 3 : Reactive Programming
JDK 8 and 9 Support JDK Changes Annotation based Programming Lambda's for Bean Registration Functional Web Framework HTTP 2 Support Server Push Servlet 4.0 Pushbuilder Library Support JUnit 5 Support	<ul> <li>@Nullable annotation</li> <li>@NotNull annotation</li> <li>Java 8 reflection enhancement</li> <li>Java 8 default methods</li> <li>Commons Logging bridge</li> <li>Resource abstraction</li> <li>Component index</li> <li>Consistent detection of transaction</li> <li>Async annotations</li> <li>GenericApplicationContext</li> </ul>	What is Reactor? Reactive Design Pattern Reactive Streams Support Event Loop Execution Model Reactive Stack Reactive Stream Adapters Spring Security Reactive Reactor and RxJava Mutable State Threading Model
Module 4 : Functional Programming	Module 5 : WebFlux	Module 6 : Kotlin and Spring
Matching Media Types Match HTTP Request Predicates Functional Endpoints? HandlerFunction RouterFunction FilterFunction Composing Functions Method References DispatchHandler Conventional Request Patterns Restful API URI Delegating Calls REST Endpoints	What is WebFlux? WebFlux Config Reactive API Annotated Controllers Spring MVC or WebFlux? Event Loop Concurrency Model Netty WebHandler API Asynchronous Service Request Handling Concurrent Requests Event Bus Implementation Mono and Flux	What is Kotlin? Functional Web endpoints Bean registration with Kotlin Constructors and Inheritance Destructuring Declarations Delegations Sealed and Data Classes Visibility Control Extension Java Interoperabiliy Kotlin's DSL Immutable Classes null Safe API

## Module 8 : Testing with JUnit 5

What is new in JUnit? JUnit Jupiter JUnit Vintage Basic Annotations @BeforeAll and @BeforeEach @DisplayName and @Disabled @AfterEach and @AfterAll Assertions and Assumptions Exception Testing Test Suites Dynamic Tests

## JVS600: Course TypeScript Programming

Code: JVS600

#### Duration: 2 days

## Audience TypeScript Programming Course

The course TypeScript Programming is intended for Web Developers who want to use Typescript and the ECMAScript 6 standard to develop the front end of modern web applications.

## Prerequisites Course TypeScript Programming

Experience with JavaScript and a good knowledge of JavaScript is required to participate in this course.

## **Realization Training TypeScript Programming**

The theory is treated with on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification TypeScript Programming**

The participants receive an official certificate TypeScript Programming after successful completion of the course.

## Course TypeScript Programming

In the course Typescript Programming participants learn to apply the TypeScript

language in modern Web applications. TypeScript is a JavaScript-based language with additional features. In particular, Typescript provides strong typing instead of the original weak typing in JavaScript. In TypeScript errors are therefore already detected during compilation instead of at run-time. The course starts with an overview of TypeScript and the features of the new ECMA 6 JavaScript standard. Next the installation of TypeScript, possible development environments and the use of various types in TypeScript is discussed. Attention is also paid to functions in TypeScript where callbacks, closures and type guards are treated. Also object oriented programming with TypeScript classes, constructors, inheritance and interfaces is part of the course program. Finally attention is given to generics, the parameterisation of types and the use of modules and namespaces in TypeScript. After completion of the course participants understand the capabilities of TypeScript which is, amongst other uses, also used in the JavaScript Framework Angular from version to onwards.

Module 1 : TypeScript Intro	Module 2 : Language Syntax	Module 3 : ECMAScript 6
What is TypeScript? TypeScript Intro TypeScript Characteristics Features of TypeScript Benefits of TypeScript Components of JavaScript Installing TypeScript TypeScript Playground Environment Setup Debugging TypeScript Typescript IDE's Visual Studio Code WebStorm Hello TypeScript	TypeScript Identifiers TypeScript Keywords TypeScript Syntax TypeScript Object Orientation TypeScript Types Handling Types Built-in Types TypeScript Variables Variable Declarations Type Assertion Variable Scope Operators Control Flow Structural Typing	Arrow Functions Enhanced Object Literals Template Strings Destructuring Spread Hoisting let Variables const Keyword Unicode Proxies Symbols Promises Reflect api Tail calls
Module 4 : Functions	Module 5 : Complex Types	Module 6 : Classes and Objects
Return Type Parameter Passing Optional Parameters Default Parameters Variable Arguments Anonymous Functions Function Constructor Recursion Lambda Functions Syntactic Variations Function Overloading Overloading Examples Specialized Overload Signature	Arrays Intro Accessing Array Elements Array Class Array Methods More Methods Functional Methods Array Destructuring Passing and Returning Arrays Tuples Tuple Operations Unions Unions and Arrays Enums	Classes in TypeScript Variables and Methods Constructors Creating Instance Variables Class Inheritance Method Overriding static Members Encapsulation Access Modifiers Interfaces Interfaces Interface Inheritance Overloading with super Factory Pattern Decorators
Module 7 : Generics	Module 8 : Modules	



Generics Explained Generic Example Generic Syntax Instantiating Generic Classes Using type T Constraining Type of T Generic Interfaces Generic Object Creation Bounded Type Parameter Runtime Type checking Generic Function Reflection Checking Object for Function Interface Checking with Generics Module Keyword Declaration File Global variables Internal Modules External Modules Working with Modules Namespaces Export and Import Module Systems Generated for AMD Generated for CommonJS Module Merging TypeScript Definiton Manager Querying for Packages

## JVS700: Course NodeJS Programming

Code: JVS700

Duration: 2 days

## Audience NodeJS Programming Course

The course NodeJS Programming is intended for Web developers that want to learn how JavaScript and the NodeJS library can be used on the server-side of Web applications.

## Prerequisites Course NodeJS Programming

Experience with JavaScript programming for the Front-End of Web Applications and a good knowledge of JavaScript is required to participate in this course.

## **Realization Training NodeJS Programming**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification NodeJS Programming**

The participants receive an official certificate NodeJS Programming after successfull completion of the course.

### **Course NodeJS Programming**

In the course NodeJS Programming participants learn how the NodeJS JavaScript



## NodeJS Programming



library can be used on the server side of Web Applications. NodeJS originated in 2009 and has since then become a powerful asynchronous development framework for creating scalable web applications. The Node is library, the popularity of which is still increasing, it is also part of the MEAN stack (Mongodb, EcmaScript, Angular and Node) which is aimed at the development of high-speed I/O-centric applications. The course starts by discussing the architecture of the Node is library with runtime modules as filesystem, events and process. Also the NPM package manager that allows community modules to be included is treated. Next the asynchronous event handling model of NodeJS is on the course schedule and the use of promises and futures of EcmaScript 6 and 7 for the handling of callbacks is explained. Also attention is paid to the processing of streaming data and File I/O in NodeJS. Also the handling of multiple concurrent client requests in NodeJS is discussed as well as the Express.js Framework that deals with templating, routing and middleware services. The Database Access module will show you how to integrate NodeJS capabilities in the backend with databases like MongoDB, MySQL and SQLite. Finally attention is paid to the creation of a Rest API and using AngularJS for Frontend.

Module 1 : Intro NodeJS	Module 2 : Event Handling	Module 3 : Streaming Data
What is Node.js? Node.js Components Node.js Applications Node.js Programming Model Important Parts of Node.js Global Objects Hello Node.js Hello Node Server Node.js Module System Finding Modules Node Package Manager Installing Modules using NPM package.json Creating a Module	Node's Event Loop Interleaving Events Timers Synchronous versus Asynchronous Blocking versus Non Blocking Using Callbacks Synchronous Mode Asynchrony with Callbacks Node Events Callback versus Events EventEmitter Class EventEmitter Class Emitting Once Nesting too Much Asynchrony with Promises	What are Streams Streams in Node.js Methods and Events Reading from a Stream Writing to a Stream Piping the Streams Chaining the Streams Piping Methods and Events What are Buffers? Creating Buffers Writing to Buffers Reading from Buffers Buffer Operations Copy and Other Methods
Module 4 : File I/O	Module 5 : Express Framework	Module 6 : Rest Services
Interacting with File System Async and Sync Async Open File Information Reading and Writing Files Closing Files Truncating and Deleting Files Creating Directories Removing Directories Reading Directories Global Objects Process Object Actions and Streams os Module	Node Web Server Node Web Client What is Express? Important Modules Hello Express Request Properties Request Methods Response Properties Response Methods Basic Routing Serving Static Files Get and Post Method File Upload Cookies Management	What is REST? REST Web Services Simple REST Examples REST Web Service Principles ID and Links Multiple Represenstation Common REST Patterns Resource URI Access Patterns JSON JSON versus XML List Users API Add User API Path Parameters Delete API
Module 7 : Database Access SQL	Module 8 : Database Access NoSQL	
MySQL Connections MySQL Queries Create Database Create and Alter Table Primary Keys Insert Multiple Get Inserted ID Select from Table Select with Filter Escaping Query Values PlaceHolders	Create Database in MongoDB Create Collection Insert into Collection Insert Multiple Documents Id Field Find One Find All Find Some Query Filter Regex	

## JVS880: Course Angular Web Development

Code: JVS880

#### Duration: 3 days

## Audience Angular Web Development Course

Web Developers who want to use the newest version of the Angular Framework libraries, version 6, in developing modern single-page Web Applications.

## **Prerequisites Course Angular Web Development**

JavaScript programming experience and a good knowledge of JavaScript is required to participate in this course.

### **Realization Training Angular Web Development**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification Angular Web Development**

The participants receive an official certificate Angular Web Development after succesfull completion of the course.

### **Course Angular Web Development**

In the Angular Web Development course, participants learn to use this JavaScript Framework to develop modern single page Web Applications. We treat Angular

version 6. The Angular Framework is a major upgrade of the AngularJS 1.x JavaScript Library, offers better performance and uses TypeScript and the new features of the ECMA 6 JavaScript standard. In the course, participants learn the architecture of an Angular application, in which a collection of components are bundled together in modules. Attention is paid to the components such as templates, annotations, views and services. Dependency injection will also be discussed. The syntax of TypeScript and the new JavaScript standard ECMA 6 is treated. The meaning of Angular CLI and the generation of artifacts as components and services and the function of the dependency injector tree to keep applications extensible and maintainable are also discussed. Further attention is paid to the new component router for basic routing, child routes and router lifecycle hooks. Forms are also on the program where template-driven forms, model-driven forms and validators are discussed. Observables open the door to reactive functional programming and their function is dismantled when making HTTP requests with the new HTTP layer. Finally participants learn to build new reusable user interface components.

Module 1 : Intro Angular 6	Module 2 : TypeScript	Module 3 : ECMA Script 6
What is Angular? New Features Angular 5 and 6 Development Environment Bootstrapping an app Displaying data Using Directives Templates Annotations Views Controllers Modules	What is TypeScript? Weak typing in JavaScript Strong Typing in TypeScript Using Built-in Types Inferred typing Explicit casting Classes Interfaces Decorators Generics Sync and await	Block Bindings Let Variables Arrow Functions Rest Parameters Spread Operator Strings Functions Closures Iterators Promises Reflection
Module 4 : Components	Module 5 : Dependency Injection	Module 6 : Routing
Component Architecture Inputs and Outputs Data flow Smart vs. Dumb components Communicating via state service Custom event bus Reusable UI Components Querying view children Querying content children Host Bindings and Listener	What is dependency injection? Writing Imports Creating services DI for component communication Configuring providers Defining provider recipes The injector tree Injecting using tokens Opaque tokens Multi Providers	Angular Routes Routing Configuration Route comparison Routing Parameters Configuring routes Linking to routes Guards Child routes Sibling routes Lazy loading routes
Module 7 : Forms	Module 8 : HTTP and Observables	Module 9 : Angular CLI
Template-driven forms Model-driven forms Tracking Changes by CSS Control ControlGroup FormBuilder Validation Validation Styling Error messages Custom validators Asynchronous validators	Performing HTTP requests Configuring request headers HttpClientModule. RxJS 6 Observables Observer Design Pattern Observables versus Promises Creating Observables Creating Subjects Emitting events Subscribing to observables Observable Operators	What is Angular CLI? Modern tooling Generate with CLI Directives and services Compile, run with CLI Deploy your applications. Generating components Generating services Transpiling TypeScript Building and serving apps Debugging



## **Angular Web Development**



## JVS900: Course ReactJS Programming

Code: JVS900

#### Duration: 3 days

## Audience ReactJS Programming Course

Developers who want to learn how to program with the ReactJS JavaScript library for building modern state of the art web applications.

### Prerequisites Course ReactJS Programming

Experience with programming in a modern programming language is required. Knowledge and experience with JavaScript is beneficial for a good understanding.

### **Realization Training ReactJS Programming**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification ReactJS Programming**

The participants receive an official certificate ReactJS Programming after successful completion of the course.

### **Course ReactJS Programming**

This course teaches how to program with the increasingly popular open-source JavaScript library ReactJS. The most recent version of ReactJS is used in the course.

React has a programming model where child component do not influence parent components in an update, providing a clean separation of components in a single-page application. After introducing the main characteristics of ReactJS such as the Virtual DOM, JSX, component renderers and callback functions, it is discussed how to create a first ReactJS app with the right combination of developer tooling like React, Webpack and Babel. ReactJS uses the functional programming style whose main characteristics in JavaScript are highlighted. Next attention is paid to React Views that put a nested component tree with HTML custom tags on the screen. The creation of components with the JavaScript extension JSX is discussed as well. Also Forms, Validation and Routing in ReactJS Applications are on the course schedule. Finally the use of the Redux Framework that is often used in combination with with ReactJS is discussed.

Module 1 : ReactJS Intro	Module 2 : Components	Module 3 : Composition and LifeCycle
What is ReactJS? ReactJS Features ReactJS is Trending Benefits of React JSX Virtual DOM What is Webpack? Webpack Module Builder Webpack Features Webpack.config ES6 Features Babel Benefits of Redux First React Script First React Component	React Components JSX Component ReactJS and JSX JSX versus JavaScript Decompiling JSX Characteristics of JSX Single Root Wrap in Parenthesis JSX XSS Protection Props Collection Props in Variables propTypes List of Props Types Default Props State Rendering State	Composing Components Refs and DOM Components with Behavior Pass State Update Accessing Child Components Mounting and Unmounting Components Lifcycle Lifecycle Details Updating Events Dynamic Tables ReadOnly ReadWrite Higher Order Components Hooks
Module 4 : Forms and Validation	Module 5 : ReactJS Routing	Module 6 : Redux Framework
Readonly ReadWrite Forms in React Controlled Components Textarea Tag Select Tag Field Level Validation Form Level Validation	Defining Routes Default Route Not Found Route Nesting and Params Multiple Params Redirects Include External Routes Running the Router Using Universal Rendering Access Router Methods	React with Redux Three Principles Redux Flow Connecting to React Store Actions Reducer Data Flow Unidirectional





## ReactJS Programming

## JVS950: Course VueJS Programming

Code: JVS950

Duration: 3 days

VueJS Programming

## Audience Course VueJS Programming

The course <u>VueJS</u> Programming is intended for Web Developers who want to use the newest version of the VueJS framework in developing modern single-page Web Applications.

## Prerequisites Course VueJS Programming

Good knowledge of JavaScript and ample <u>JavaScript</u> programming experience and is required to participate in this course.

## **Realization Training VueJS Programming**

The theory is treated on the basis of presentations. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification VueJS Programming**

The participants receive an official certificate VueJS Programming after succesfull completion of the course.

## **Course VueJS Programming**

In the course VueJS Programming participants learn to use the VueJS JavaScript

Framework to develop modern single page Web Applications. VueJS is a popular JavaScript Framework that is experiencing rapid growth due to its simplicity, good performance and small footprint. VueJS started as part of the ecosystem of the well-known Laravel PHP Framework, but is now also widely used outside of it. The integration of VueJS into an application can start with a simple script tag and can then grow with the need from 3 lines of code to managing the entire view layer. Knowledge of Babel or Webpack is not necessary to start with VueJS, but VueJS can work together with these tools. The course VueJS programming starts with a discussion of the installation of the Framework and important components such as the root Vue instance, templates, data and directives, two way data binding, watchers and filters. Subsequently templates, virtual DOM and binding directives are discussed. The use of components, events, props, validation and named and custom slots are also treated. Styling with CSS is on the course program as well as rendering with JSX. Next attention is paid to routing in VueJS with nested, dynamic and programmatic routing. Asynchronous processing with promises and state management are also discussed. Finally attention is paid to VueJS. Tooling and the internals of VueJS.

Module 1 : VueJS Intro	Module 2 : Templates	Module 3 : Components
What is Vue.js	HTML-Based Syntax	Component Basics
Installation and Setup	Virtual DOM	Component Registration
vue-loader and webpack	Render Functions	Props
root Vue Instance	Interpolations	Data and Methods
Options Object	Mustache Syntax	Computed Properties
Templates, Data and Directives	v-once Directive	Prop Validation
Looping in Templates	Raw HTML	Data Flow
Binding Arguments	v-html Directive	.sync Modifier
Reactivity	Attributes	Passing Content with Slots
Two Way Data Binding	v-bind Directive	Named and Scoped Slots
Instance Life Cycle Hooks	JavaScript Expressions	Custom Events
Computed Properties	Arguments	Mixins
Watchers	Modifiers	Non-prop Attributes
Filters	v-bind and v-on Shorthand	Components and v-for
Module 4 : Styling	Module 5 : Rendering with JSX	Module 6 : Routing
Class Binding	What is JSX?	HTML5 History Mode
Object Syntax	Render Function	Dynamic Routing
Inline Style Binding	Nodes, Trees and Virtual DOM	Passing Parameters
Array Syntax	Virtual Nodes	Nested Routes
Multiple Values	Tag Name	Redirect and Alias
Auto-prefixing	Data Object In-Depth	Navigation
Multiple Values	createElement Arguments	Active Class
Scoped CSS with vue-loader	Constraints	Programmatic Navigation
CSS Modules with vue-loader	Event and Key Modifiers	Navigation Guards
Conditional Rendering	Functional Components	Route Order
Preprocessors	Template Compilation	Route Names
Module 7 : State Management	Module 8 : Vue Tooling	Module 9 : Vue Internals
State Concept	Example Sandbox	Reactivity in Depth
Vuex	WebPack and NPM	Object.defineProperty
State Helpers	Component Scoped CSS	Dependency Tracking
Mutations	Vue Test Utils	Change Notification
Synchronicity	Jest and Mocha	Tracking Changes
Actions	Asynchronous Updates	Watchers and Rerendering
Action Helpers	TypeScript Support	Property Addition nor Detected
Destructuring	Plugins	Touch and Notify
Promises	Annotating Return Types	Change Detections Caveats
Modules	Browserify	Declaring Reactive Properties
File Structure	Pre Compiled Templates	Async Update Queue

## JVS990: Course Polymer Web Development

Code: JVS990

Duration: 3 days

## **Audience Polymer Web Development**

The course **Polymer Web Development** is intended for Web Developers who want to use the newest version of the Polymer JavaScript Framework in developing elements for modern single-page Web Applications.

### **Prerequisites Course Polymer Web Development**

A good knowledge of JavaScript and <u>JavaScript programming</u> experience and is required to participate in the Polymer Web Development course.

## **Realization Training Polymer Web Development**

The theory is treated on the basis of presentations. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

## **Certification Polymer Web Development**

The participants receive an official certificate Polymer Web Development after succesfull completion of the course.

### **Course Polymer Web Development**

In the course Polymer Web Development participants learn to use the Polymer





## **Polymer Element Development**





JavaScript Framework when developing modern single page Web Applications. Polymer is an open source JavaScript library for building web applications based on Web Components. Modern design principles are implemented in a separate Material Design project. Polymer is not a complete framework for building web applications such as Angular, but instead emphasizes reusable Web Components. While Angular has APIs for things like services, routing and server communication, Polymer offers these things as Web Components in the core library. These Web Components are not bound to Polymer but can be used like lego blocks in other web application frameworks. The course starts with the installation of Polymer and with an intro about the fundamental concepts that play a role in Web Component specifications such as HTML Templates, HTMLImports, Shadow and Shady DOM. Subsequently the development of Custom Elements in Polymer, data binding and the implementation of life cycle callbacks are discussed. The various Material Design custom elements that Polymer makes available as standard are also addressed. There is ample attention for reuse and styling with CSS where DOM manipulation, encapsulation of style and shims are discussed. Finally event handling, Polymer tooling and application building are treated.

Module 1 : Polymer Intro	Module 2 : Polymer Elements	Module 3 : Properties and Databinding
What is Polymer? Web App Complexity What are Web Components? Web Component Specifications HTML Templates Activating Templates HTMLImports Shadow and Shady DOM Downloading Polymer Polymer Framework Files Installing Polymer Polymer Directories	Custom Elements Creating and Extending Polymer Custom Elements Importing HTML Hello Polymer Revisited Including Hello Element Local DOM Element Sections Class Style Constructor Type Extension Element Lifecycle Callbacks Attaching and Detaching	Element Properties Property Key Value Property Types Data Binding Objects Native Element Binding Attribute Binding Computed Properties Observers Observer Array Mutations Conditional Templates Template Repeater and Array Selector Autobinding Template
Module 4 : Polymer Element Catalog	Module 5 : Reuse and Styling	Module 6 : Events and Behavior
Material Design Iron Elements Iron-Ajax-Element Paper Elements Paper Input Element Google Web Components Gold Elements Gold cc input element Neon Elements Platinum Elements Platinum Service Worker App Elements Routing with App-Route	Local DOM Insertion Points Encapsulation and Scope Shadow and hady DOM Light DOM Accessing the DOM Polymer DOM API Observe Node Changes In and Outside CSS Styling CSS Variables and Mixins Custom Property API Shim Limitations Custom and Shared Styles	Polymer Events Event Listeners on-event Annotations Add and Remove Listeners Listener on Custom Element Listener on Outside Elements Custom Events Gesture Events Gesture Event Types Event Retargeting Property Changed Events Behaviors
Module 7 : Polymer Tooling	Module 8 : Building Applications	
Tools Overview Polymer CLI CLI Commands Polyfills ES6 Modules Create Element Project Create Application Project Element Documentation Testing Elements polymer.json Specification Node Support Using NPM	App Templates app-layout Elements Build for Production polymer-build Build Configuration HTTP/2 Push Pre Cache Lazy Load Multiple Builds Server Apps PRPL Pattern Service Workers	

## **MOB100: Course Android Fundamentals**

Code: MOB100

#### Duration: 1 day

Price: € 499

## **Audience Android Fundamentals Course**

This course is intended for developers, managers and others who want to get an overview of the capabilities and operation of the Android platform.

## **Prerequisites Course Android Fundamentals**

To participate in this course knowledge of the basics of Java programming is required.

### **Realization Training Android Fundamentals**

The theory is discussed on the basis of presentation slides. Demos are used to clarify the concepts. The theory is interspersed with practical exercises. The course material is in English.

## **Course Android Fundamentals**

In the course Android Fundamentals participants learn the fundamentals of the Android platform and how to develop apps for Android. Next it is discussed how a development environment for Android can be installed and arranged under Eclipse. Attention is paid in this respect to the Android SDK, the Eclipse IDE plugin and device emulators. The function and use of XML resource files is discussed as well as the inplementation of the user interface. An important element of the course is the life cycle of processes or activities. And further it is discussed how to debug Android apps, how to access databases and how to use threads and content providers. Finally the





## Android Fundamentals





participants learn how to publish their app on Google Play. With the knowledge gained in this course, participants will be able to write their own simple apps and they will understand the architecture and operation of the Android platform.

Module 1 : Android Intro	Module 2 : Android Development	Module 3 : Android GUI
The Basics What is Android? Architecture Explained Role of Java Android SDK Eclipse IDE Plugin Device Emulator Profiling Tools Hello World Application	Unit Testing and Debugging Creating Unit Tests Android Development Tools (ADT) Using the Emulator TraceView Analyzing the Heap	Android User Interface View Hierarchy Menus Fragments Styling Widgets Defining Styles Applying Styles to the UI Platform Styles and Themes Layout Layout Containers Weight and Gravity Layout Techniques Handling Events Advanced Widgets ScrollView ViewPager TabView Custom Views
Module 4 : Android Persistence	Module 5 : Android Threads	Module 6 : Android Service
Storing and Retrieving Data Internal and External Storage Preferences SQLite Database Content Providers Querying Content Providers Modifying Data Creating a Content Provider	Asynchronous Tasks Main UI Thread Using AsyncTask Location Services and Maps Location Services Mock Location Data Google Map Libraries	Accessing Remote Services HTTP DOM Parsing SAX Parsing JSON Parsing Application Fundamentals Alarm Manager Broadcast Receivers Services Notification Manager

## Module 7 : Best Practices

Web Apps Overview Targeting Screens from Web Apps WebView Debugging Web Apps Best Practices for Web Apps Compatibility Supporting multiple screens Optimizing for Other Android Versions

## **MOB200: Course Android Programming**

Code: MOB200

#### Duration: 3 days

## Audience Android Programming Course

This course is intended for experienced Java developers who want to learn how apps can be programmed for Android.

### Prerequisites Android Programming Course

To participate in this course knowledge of and experience with programming in Java is required.

## **Realization Training Android Programming**

The theory is discussed on the basis of presentation slides. Demos provide an illustrative clarification of the discussed concepts. The theory is interspersed with exercises. The course material is in English.

## **Certification Android Programming**

Participants receive an official certificate Android Programming after successful completion of the course.

### **Course Android Programming**

In the course Android Programming participants learn app development for the Android operating system. First a global overview of the Android platform is given and

it is explained what makes Android unique and how it fundamentally differs from other platforms. Then the main building blocks of Android apps are on the agenda such as Activities and User Interface. Also Files and Preferences will get attention. During the course the participants develop an app that has a nice user interface, which uses web services for accessing cloud applications and which has a database for local data storage. The app is made with security in mind and works on any form factor from smartphone to tablet and TV. During the course, participants learn when and how the main building blocks of Android app development can be used. The participants come into contact with many facets of app development for Android such as activities, services, providers and receivers. Also best practices of Android development and debugging and testing apps is discussed. The latest version of the Android OS is used in the course with Android Studio as IDE and emulators for devices.

Module 1 : Android Overview	Module 2 : Main Building Blocks	Module 3 : Activities and User Interface
History of Android Android Stack Android Architecture Dissecting Android apps Building blocks Debugging Testing Android Security Creating your first project The manifest file Layout resource Running your app on Emulator	Activities Activity lifecycle Intents Services Content Providers Broadcast Receivers	Understand the Lifecycle Callbacks Specify Your App's Launcher Activity Create a New Instance Destroy the Activity Pause Your Activity Resume Your Activity Stop Your Activity Stop Your Activity Save Your Activity State Restore Your Activity State XML versus Java UI Dips and sps Views and Iayouts Common UI components Handling user events
Module 4 : Preferences and Files	Module 5 : Advanced UI	Module 6 : Device Support
Get a Handle to a SharedPreferences Write to Shared Preferences Read from Shared Preferences Choose Internal or External Storage Obtain Permissions for External Storage Save a File on Internal Storage Save a File on External Storage Query Free Space Delete a File	Support Libraries Selection components Adapters Complex UI components Building UI for performance Menus and Dialogs Graphics and animations	Create Locale Directories and String Files Use the String Resources Create Different Layouts Create Different Bitmaps Specify Minimum and Target API Levels Check System Version at Runtime Use Platform Styles and Themes
Module 7 : SQL Database	Module 8 : Content Providers	Module 9 : Multimedia in Android
Introducing SQLite SQLiteOpenHelper and creating a database Opening and closing a database Working with cursors Inserts, updates, and deletes	Content provider MIME types Searching for content Adding, changing, and removing content Working with content files	Multimedia Supported audio formats Simple media playback Supported video formats Simple video playback






## **MOB400: Course iPhone Programming**

Code: MOB400

#### Duration: 3 days

### Audience iPhone Programming Course

This course is intended for developers who want to learn to create apps for the iPhone and the iPad.

### Prerequisites Course iPhone Programming

To participate in this course it is desirable to have some knowledge of Objective-C.

### **Realization Training iPhone Programming**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Demos provide a clarification of concepts. The course materials are in English.

### **Course iPhone Programming**

In the course iPhone Programming participants learn to use Objective-C and the Foundation Framework to develop apps for the iPhone and iPad. The course uses the latest version of Objective-C. Participants will gain experience with Xcode and Interface Builder. The syntax of Objective-C will be discussed and participants will learn how user input from iPhone and iPad can be collected and processed in the app. Furthermore participants will learn how user interfaces are built using iPhone and iPad controls such as text fields, buttons and pickers. Also attention is paid to how apps can use iPhone table views, navigation controllers and other UI components.





### **IPhone** Programming





Finally the focus of the course is on how you can save the state in an iPhone or iPad app and how you can store user data.

Module 1 : XCode Intro	Module 2 : Objective C	Module 3 : Interface Builder
What is XCode Using Xcode Editing XCode Build and debug iPhone apps Managing project resources	Objective-C programming language Variables Data types Control Flow Error Handling Foundation Framework	Overview of Interface Builder GUI tool Designing your iPhone app's UI Connect the UI to your code File's Owner First Responder
Module 4 : Cocoa Touch	Module 5 : Table Views	Module 6 : Navigation
Cocoa Touch Design Patterns Model-View-Controller Other design patterns iPhone's UIKit framework	UITableView UITableViewController Implementing data source methods Implement delegate methods Present data Interact with the user	iPhone's navigation controller Tab bar controller Other UlKit components iPad-specific controls popovers
Module 7 : Touch Events	Module 8 : Custom Drawing	
Standard iPhone controls Standard iPad UI controls Managing custom components Response to user input	Core Graphics Core Animations Customise iPhone iPad UI components Displaying images Manipulating images Rich animation Graphical elements	

## **MOB500: Course Xamarin Fundamentals**

Code: MOB500

Duration: 2 days

### **Audience Xamarin Fundamentals**

This course is intended for developers who want to use the Xamarin Framework for the creation of cross-platform mobile apps for iOS, Android and Windows.

#### **Prerequisites Course Xamarin Fundamentals**

To participate in this course knowledge and experience with programming in C# or Java is required. Experience with Visual Studio and Mono for Android is desirable.

#### **Realization Training Xamarin Fundamentals**

The theory is discussed on the basis of presentation slides and is interspersed with exercises. Illustrative demos are used to clarify the concepts. The course material is in English.

### **Course Xamarin Fundamentals**

In the course Xamarin Fundamentals participants learn to create apps with the Xamarin Framework. The Xamarin 2.0 Framework allows you to build mobile Apps for both iPad and iPhone devices as well as for Android and Windows devices. The Visual Studio .NET development environment is used with C# as programming language. Furthermore also Mono for Android is used which has become an excellent development platform for creating apps for Android. After an introduction where the setup the Xamarin development environment is discussed, general characteristics of





## Xamarin Fundamentals





cross-platform development like the use of portable class libraries are treated. Attention is paid to the creation of User Interfaces for apps in iOS and Android where Xamarin Forms play an important role. Working with Activities in Android and View Controllers in iOS is also discussed. Then the focus moves to how you can deal in Xamarin with different data formats like XML and JSON, how to use database functionality and REST Web Services. Finally, attention is also paid to the use of location services and the camera. During the course iOS Apps and Android Apps are built with the use of Xamarin and Visual Studio.

Module 1 : Xamarin Intro	Module 2 : Cross Platform Development	Module 3 : User Interfaces
Cross Platform Development Xamarin background Xamarin tools Installation Device emulators GenyMotion virtual Android devices XCODE iPad/IPhone emulators Mono, MonoTouch and Mono.Android Visual Studio Integration	Cross-platform development Local storage App navigation techniques Native or HTML Xamarin.Forms Visual Studio Setup Linked files in Visual Studio. Shared Projects Compiler directives Portable Class libraries	AXML markup Android coded UI approach iOS Interface Builder iOS coded UI MonoTouch Dialog framework HTML WebView control HTML UIWebView control Xamarin.Forms in detail
Module 4 : Activities and View Controllers	Module 5 : Using REST	Module 6 : Data Access
Android Activities Activities lifecycle ListActivity ListAdapter iOS Controllers iOS ViewController lifecycle iOS NavigationController. UITableViewController UITableViewSourceIntroduction to REST HTTP call JSON serialisation	Public REST API. Data contracts Building proxies Asynchronous patterns Async en await REST in Android REST in iOSXML access JSON access	Using SQLite in Android Using SQLite in iOS SQLite and ORM Using external storage Synchronisation to cloud

### Module 7 : Using Location

Securing dataAndroid Location manager. iOS location manager classes. Using maps op iOS Using maps on Android Translate location coordinates Tracking applications Using the camera

# MOB550: Course Xamarin Mobile Development

Code: MOB550

### Duration: 4 days

Module 1 : Xamarin Intro	Module 2 : App Structure	Module 3 : Text Handling
Cross Platform Development Xamarin background Xamarin tools Installation Device Emulators GenyMotion virtual Android devices XCODE iPad/IPhone emulators Mono, MonoTouch and Mono.Android Visual Studio Integration	Cross-platform development Local Storage App Navigation Techniques Native or HTML Xamarin.Forms Linked files in Visual Studio. Shared Projects Compiler directives Portable Class libraries	XAML markup Android coded UI approach iOS Interface Builder iOS coded UI MonoTouch Dialog Framework HTML WebView Control HTML UIWebView Control Xamarin.Forms in Detail Form Validation
Module 4 : Layouts	Module 5 : Event Handling	Module 6 : Data and Location Access
Android Activities Activities lifecycle ListActivity ListAdapter iOS Controllers iOS ViewController lifecycle iOS NavigationController. UITableViewController UITableViewSourceIntroduction to REST HTTP Calls JSON Serialisation	Public REST API Resource ID's Standard Methods Data Contracts Building proxies Asynchronous patterns Async and await REST in Android REST in iOS XML Access JSON Access	Using SQLite in Android Using SQLite in iOS SQLite and ORM Using External Storage Synchronisation to Cloud iOS location Manager Classes Using maps op iOS Using maps on Android Translate Location Coordinates Tracking Applications Using the camera
Module 7 : Data Binding	Module 8 : MVVM Architecture	Module 9 : Collection Views
Bindable Infrastructure Bindable Object Class BindableProperty and BindingBase LINQ Expression Object SetBinding method BindingExtension class ReferenceExtension class INotifyPropertyChanged IValueConverter x:Reference binding Binding Markup Extension Binding Context Binding Context Propagation Binding Modes	MVVM Layers Model for Data View for UI ViewModel Connection Layer Communication Interactive properties Xamarin.FormsBook.Toolkit Color ViewModel Streamlining View Model Command Interface Execute CanExecute Simple Method Executions Application Lifecycle	Collection Types IEnumerable and ICollection Picker Options with Picker Picker Data Binding SelectedIndex ListView ItemsSource Collections and Selections Row Separator ObservableCollection ListView Interactivity TableView Templates and Cells
Module 9 : Triggers and Behaviors	Module 10 : Page Navigation	
Trigger Objects PropertyChanged Handler pre-Trigger value Trigger Actions Trigger Animations EventTriggers ScaleUpAndDownAction Data Triggers and MultiTriggers Combining Conditions AndConditions Behaviors Behaviors Behaviors Responding to Taps	Start Pages Back Button Navigation Modal Pages Modeless Pages PushAsync PushModelAsync Navigation Bar Task Objects PopAsync PopModalAsync Enforcing Modality Navigation Variations Navigation Menu Dynamic Page Generation	

## MOB600: Course PhoneGap Programming

Code: MOB600

#### Duration: 2 days

### Audience PhoneGap Programming Course

This course is designed for web developers, mobile developers and designers and project managers in

mobile projects who want to learn about cross-platform programming with PhoneGap.

### Prerequisites Course PhoneGap Programming

Knowledge and experience with HTML, CSS and JavaScript and the mobile ecosystem is required to participate in this course.

### **Realization Training PhoneGap Programming**

The subjects are discussed on the basis of presentation slides. Demos help to clarify the concepts further. The theory is interspersed with exercises. Mobile devices are emulated in the Eclipse IDE. The course material is in English.

#### **Course PhoneGap Programming**

In this course the participants learn to create cross-platform mobile applications using the open-source framework PhoneGap. It is discussed how PhoneGap allows you to build mobile apps while adding re-use to your code and other assets using well known web technologies like HTML, CSS and JavaScript. Attention is paid to setting up the development environment to compile for multiple platforms and to run and debug the HTML and JavaScript. It is discussed PhoneGap Build can be used to

create app-store-ready applications for iOS, Android, and Windows Phone. Also deployment and installation of the application on device on like a smart phone or table is treated. Also the optimization of JavaScript for mobile devices, the use of CSS transitions, animations and transforms to create native looking interfaces and the use of mobile JS libraries, such as XUI and Dashcode to build mobile applications faster, are on the course schedule. Finally is is discussed how to access native APIs, including location, camera, accelerometer, contacts, and more via JavaScript.

Module 1 : Cross Platform Development	Module 2 : PhoneGap Essentials	Module 3 : PhoneGap Build
What is Phonegap? Cross-platform development jQuery jQuery Mobile AJAX and JSON Local storage App navigation techniques Introducing PhoneGap Native or HTML	PhoneGap Walkthrough Apache Cordova Adobe PhoneGap Build Accounts Page Model Developing in the Browser Writing PhoneGap-Only Code Initialization Using PhoneGap APIs Detecting Devices Navigation Responsive Design	What is PhoneGap Build Eclipse with Plug-Ins Running Android Simulators Building your first App Deploying on an Android What is Required? Viewing Console with Android SDK What about Icenium? Optional Apple Mac setup in XCode Why not use PhoneGap Build?
Module 4 : Mobile Projects	Module 5 : Developing for iOS	Module 6 : Developing for Android
Mobile Projects Creating a new project Importing a boiler-plate project Enabling PhoneGap for existing projects What are Signing Certificates? What is Provisioning? Obtaining and Using Signing Certificate Keys	Workflow of iOS Development Apple Developer program Getting a Developer Account Preparing the Mac Provisioning a Device Building a Development IPA Building a Distribution IPA Submitting to the AppStore Signup for Google Play	Android Developer keys Register Your App Signing Your App Upload to Google Play Kindle Android Store Registering with the Store Testing Your App Uploading for Certification Why not PhoneGap Build? Building the Project
Module 7 : Developing for Windows	Module 8 : Mobile Coding Techniques	Module 9 : Miscellaneous Topics
Windows Phone 8 emulators Windows keys Ripple Emulator in Chrome Free Microsoft tools for Windows 8 Adding a Privacy Policy Working with WinJS Code Preparing for the Store Submitting to the Store	Mobile Coding Techniques Writing Optimised Javascript Managing memory on mobiles CSS optimisations User-Interface best practices	Accessing Native APIs When to go native Geolocation Camera Accelerometer Contacts Other device-specific APIs





## **MOB700: Course Objective C Programming**

Code: MOB700

#### Duration: 3 days

Price: € 1400

### Audience Objective C Programming Course

This course is designed for participants who want to learn the basics of Objective C in preparation for the development of iOS and OS X applications.

#### Prerequisites Course Objective C Programming

No programming knowledge is required to participate in this course. Prior knowledge of programming languages such as Java or Visual Basic is beneficial for the understanding.

### **Realization Training Objective C Programming**

The theory is treated on the basis of presentation slides. Demos are used to clarify the theory. There is ample opportunity to practice and theory and exercise are interspersed. The course uses the XCode development environment.

### **Course Objective C Programming**

In the course Objective C Fundamentals participants learn the basics of the Objective C programming language for iOS development. The course does not address the development of iOS app, but is a preparatory course in which the Objective C language in which iOS apps are written is treated. First Apple's XCode development

environment is discussed and how to create and manage projects. Next attention is





## **Objective C Fundamentals**





paid to the basic syntax of the ANSI C programming language with statements, variables, data types, declarations, selections with if and iterations with for and while loops. Also more advanced features of C are addressed like using functions and passing of parameters, creating user defined data types with structs and typedefs, working with pointers and addressing and allocating memory. Next specific Objective C features like objects, classes, instance variables and methods are discussed as well as object communication through messages, accessors and properties. Also the Foundation framework that provides a set of useful primitive Objective C classes is treated as well as the functionality that it adds to Objective C. Finally the memory management model of Objective C is on the course program.

Module 1 : XCode Intro	Module 2 : ANSI C language	Module 3 : C constructs
Xcode IDE Project creation Setting preferences Using navigator Groups Code Editor Jump bar Utilities	Variables Data types C library Operators Selections if and else switch and case Iterations for and while	Arrays Multiple dimensions Functions Scope of variables Parameter passing Call by value Structs Typedefs Enums Pointers
Module 4 : Objective C	Module 5 : Foundation Framework	Module 6 : Memory Management
Objects and Classes Declaring a Class Implementing a Class Creating Objects Instance variables Using Methods Accessors Using Properties Property Attributes Conventions Messages Method Dispatch Initializers Custom Initializers Designated Initializers Initializer Chain	What is Foundation Framework NSObject Strings NSString Using Arrays NSArray NSNumber NSDictionary NSError NSData NSException	Understanding Memory Understanding Memory Address Basics Memory Pointers Addresses Understanding Pointers Stack and Heap Memory Automatic Memory Management Requesting Memory Deallocating Memory Managing Memory in Objective-C Using the Retain/Release Model Retain counts

## **MOB800: Course Swift Programming**

Code: MOB800

Duration: 3 days

### Audience Swift Programming Course

The course Swift Programming is designed for participants who want to learn the basics of the newest version of Apple's programming language Swift for iOS apps.

### Prerequisites Course Swift Programming

No programming knowledge is required to participate in this course. Prior knowledge of other programming languages such as Objective C, Java or JavaScript is beneficial for the understanding.

### **Realization Training Swift Programming**

The theory is treated on the basis of presentations. Demos are used to explain the theory. There is ample opportunity to practice and theory and exercises are interspersed. The course uses the newest version of the XCode development environment.

### **Course Swift Programming**

In the course Swift Programming participants learn the basics of this programming language for developing iOS apps. After an introduction into the XCode development environment, playgound projects and iOS projects, the syntax of the language is discussed. Also the relationship with Objective C is explained. Attention is paid to Swift versions of well known C data types like Int for integers and Float for floating



Think different

## Swift Fundamentals





point. Also the main Collection types, Array and Dictionary, are on the course program. Swift uses a lot of variables that may not change value and this makes the code safer and clearer. The benefit and use of these constants will be given special attention. The new advanced types like tuples are introduced as well. After the treatment of functions and parameter passing, classes with properties and methods are discussed. Unlike some other languages Swift does not require separate interface and implementation files for classes. Classes are defined in a single file and are then available for calling code. Finally closures are treated which can be regarded as pieces of functionality that can be passed around and used.

Module 1 : Swift Intro	Module 2 : Swift Syntax	Module 3 : Swift Programming
What is Swift? Why Swift Comparison with Objective C XCode Environment Environment setup Creating Playground Project Creating iOS Project .playground files Setting preferences Using navigator	Constants Variables Strings Interpolation Statements Printing Comments Data Types Integers and Floats Tuples Enumerations	Operators Arithmetic Operators Comparison Operators Conditionals Overflow Checking Looping XCode Playground Timeline Array Mutability Array Iterations Dictionaries Mutability of Dictionaries Using Dictionaries Tuples
Module 4 : Functions	Module 5 : Classes and Objects	Module 6 : Closures
Code reuse with Functions Defining Functions Calling Functions Parameters Scope of Declarations External Parameter Names Default Parameter Values Returning tuples Nested Functions Recursion	What are Classes? Class Definition Classes and Objects Access Modifiers Class Methods Properties Attributes Initializers Value Types Reference Types Method Overloading Inheritance	Closure intro Closure Expressions Internal Iteration Mapping and Reducing Computed Properties Optionals Optional Values Optional Binding Variadic parameters

## MOB850: Course iOS Development with Swift

Code: MOB850

#### Duration: 5 days

### Audience iOS Development with Swift Course

This course is intended for developers who want to use the Swift programming language to develop apps for the iPhone and the iPad.

#### Prerequisites Course iOS Development with Swift

Some knowledge of programming in Objective C or Swift is required to participate in this course.

#### **Realization Training iOS Development with Swift**

The theory is treated on the basis of presentation slides. Demos are used to explain the theory. There is ample opportunity to practice and theory and exercises are interspersed. The course uses the latest version of iOS SDK, Xcode and Interface Builder.

### **Certification iOS Development with Swift**

Participants receive an official certificate iOS Development with Swift after successful completion of the course.

#### **Course iOS Development with Swift**



X iOS

## iOS Development with Swift





In the course iOS development with Swift participants learn to use the programming language Swift for developing apps for the iPhone and the iPad. After an introduction into the XCode IDE and a review of the Swift programming language attention is paid to the anatomy of an iOS application, the sandbox environment and the characteristic architecture of iOS apps namely the model view controller architecture. The participants learn to work with the XCode IDE, Interface Builder, the UIKit Framework and the iOS SDK. During the course several apps are developed such as a newsreader app and apps that make use of multiple views. These apps use various controls such as buttons, switches, tables and maps. Also features specific to iPad apps such as split views and popup menus are addressed. Further attention is paid to multitasking, HTTP networking, web service access, data access with Core Data and SQLite and dealing with XML and JSON data. Participants learn to use storyboards for UI design and examine drawing techniques and animation. During the course also the debugging of iPhone and iPad apps are discussed. The course concludes by adressing deployment issues and localization of apps.

Module 1 : Apps Intro	Module 2 : Swift Overview	Module 3 : App Architecture
XCode IDe Creating projects Templates, Projects, and Workspaces Creating a New Project LLVM and LLDB Debug Gauges Asset Management XCTest Testing Framework Anatomy of an iOS Device iOS Architecture Available SDKs Version Compatibility	Statements Constants Variables Data Types Collection Types Functions Closures Classes and Structures Automatic Reference Counting (ARC) Optionals Protocols Generics Objective-C Interoperability	MVC architecture Model View Controller IBOutlets IBActions Subclassing Delegation Root View Controller Navigation Controller Controlling Stack Navigation
Module 4 : UIControls	Module 5 : Views	Module 6 : Multitasking
Interface Builder Basic Interaction Buttons Text Fields Sliders Segments Switches Action sheets Alerts Scrolling Image scrolling Zooming images Paging images Data Picker Hiding the Keyboard	What are views The View Hierarchy View behavior Containers Controls Dynamic applications Swap views Collection views Grids Text and Web Views Navigation View Tab Bars Alert Views and Action Sheets Display dynamic web pages	Application States Background Execution Background App Refresh State Restoration Concurrency Grand Central Dispatch (GCD) Serial and Concurrent Queues Main Dispatch Queue Completion Blocks Operation Queues
Module 7 : Advanced Controls	Module 8 : Persistence	Module 9 : RSS and JSON

Pickers Tables Customizing Tables Static Table Views Dynamic Table Views Delegates DataSources Table View Styles Custom Cells Tab Bars Using MapKit Display Maps Navigate Locations Drop Pins	Storing user preferences NSFileManager NSFileHandle NSData Pathnames in Swift Working with Directories Working with Files Reading and Writing from a File Key-Value Data SQLite Integration Using SQLite Directly Overview of Core Data Managed Objects Persistent Store Coordinator Entity Descriptions Retrieving and Modifying Data	Reachability Synchronous Downloads Asynchronous Downloads Handling Timeouts Sending HTTP GET Requests Sending HTTP POST Requests Using RSS Using JSON Parsing JSON Parsing XML AirDrop
Module 10 : Layouts and Storyboards	Module 11 : Drawing and Animations	Module 12 : Advanced Topics
Auto layout View Autosizing Handling screen size Handle rotation Size classes Split view controllers Controlling Rotation Behavior What are Storyboards? Adding Scenes Segues Transitions Using in a Tab Bar Application	The Responder Chain Touch Notification Methods Scheduling notifications Respond to notifications Enabling Multitouch on the View Gesture Motions Gesture Recognizers Drawing Core Graphics and Quartz 2D Lines, Paths, andShapes Animation Core Animation Blocks Animation Curves Transformations SpriteKit SceneKit Physics Engine Adding Effects	Targeting Multiple Devices iPhone vs. iPad Universal Apps Detecting Device Capabilities Supporting Multiple iOS Versions Handoff Interactions App Framework Support Implementing Handoff Continuation Streams App Extensions Localization Resources Running on a Physical Device Development Certificates Assigning Devices Creating an App ID Provisioning Profiles

## MOB900: Course iOS Development with Objective C

Code: MOB900

Duration: 5 days

### Audience iOS Development with Objective C Course

This course is intended for developers who want to use Objective C for developing iOS apps for the iPhone and the iPad.

#### Prerequisites Course iOS Development with Objective C

To participate in this course some knowledge of and experience with programming in Objective C is required.

#### **Realization Training Development with Objective C**

The theory is treated on the basis of presentation slides. Demos are used to clarify the theory. There is ample opportunity to practice and theory and exercises are interspersed. The course uses the latest iOS SDK 8, 6 Xcode and Interface Builder.

### **Course iOS Development with Objective C**

In the course iOS Development with Objective C, participants learn to develop apps for the iPhone and iPad with the programming language Objective C and Apple's Cocoa Touch framework. Attention is paid to working with XCode 6 and other development tools and the iOS application development architecture. After a review of the Objective C language the use of appropriate techniques and patterns in Objective C is discussed. Next the design of User Interfaces with UI controls and storyboards is discussed and techniques for drawing and animation are investigated. The use of





## iOS Development with Objective C





multitasking with threads in iOS and running apps in the background is treated. In mobile device there is a need to store data locally as user data or preferences and this is shown on the basis of Core Data and SQLite. Additionally the processing of XML and JSON data in apps is on the program and also a number of issues that have to do with deploying apps on devices. Finally best practices to program for both iPhone and iPad devices are discussed.

Module 1 : iOS Intro	Module 2 : Objective C Overview	Module 3 : MVC Architecture
Versions of iOS Supported Devices Command Line Interaction Apple Approval Process Installing XCode Updating XCode Code Development Code Tracking Debugging Utilities Navigation Tools Anatomy of an iOS Device iOS Architecture Available SDKs Version Compatibility	Basic Syntax Objects and Classes Using Dynamic Binding Using ARC Hardware Interaction Data Types and Structs Flow Control Pointers Heap and Stack id Data Type Type Safety Function Support of Objects Protocols Categories Blocks	MVC Paradigm Role of Model Role of View Role of Controller Application Design Process Model View Controller (MVC) Target-Action Pattern Subclassing Delegation Service Oriented Architecture Web Services Overview Apple Push Notification Service (APNs)
Module 4 : Using NextStep Types	Module 5 : Views	Module 6 : Multitasking
NSDate NSPoint NSSize NSString Collection Objects NSSet NSMutableSet NSArray NSMutableArray NSMutableArray NSMutableDictionary	Views and View Controllers Using Generic Views View Controllers Apple's UI Design Standards Utilizing Controls Storyboards Specific Purpose Views Text View Web View Map View Table and Table Cell View Designing Views Custom View Transitions Custom Animations	Creating Threads Asynchronous Processing Syntax of Blocks Purpose of Blocks Block Animation Methods Multithreaded Blocks Apps Moving to Background Detecting Low Memory Events Solving Low Memory Events Handling Custom Events Handling Exceptions
Module 7 : Persistence	Module 8 : Networking	Module 9 : Deployment
Storing user preferences NSFileManager NSFileHandle NSData Pathnames in Swift Working with Directories Working with Files Reading and Writing from a File Key-Value Data SQLite Integration Using SQLite Directly Overview of Core Data Managed Objects Persistent Store Coordinator Entity Descriptions Retrieving and Modifying Data	Reachability Synchronous Downloads Asynchronous Downloads Handling Timeouts Sending HTTP GET Sending POST Requests Parsing JSON Parsing XML Validating XML Transforming XML	Signing Assets Developer Certificates Assigning Devices Xcode Integration Performance Power Optimization Measuring Performance Responsiveness Memory Usage Networking and Power Deployment Icons and Launch Images Archiving Distribution iTunes Connect

## **NET100: Course .NET Overview**

### Code: NET100

Duration: 1 day

**Price:** € 499

### Audience Course .NET Overview

This course is designed for people who want to get an overview of the features and functions of the .NET framework.

### **Prerequisites Course .NET Overview**

Knowledge of software development and some programming knowledge is required to participate in this course.

#### **Realization Training .NET Overview**

The concepts are discussed on the basis of presentation slides. The concepts discussed are demonstrated with demos in Visual Studio.NET.

### **Course .NET Overview**

This course provides an overview of the operation and capabilities of the .NET platform and how the way to develop applications for Windows and the Web has changed with the advent of .NET. Attention is paid to the advantages that .NET provides for the scalability and flexibility of applications and the how and why of the Common Language Runtime. The different types of applications are discussed in .NET, such as Windows Forms applications, Web Forms applications and Web Services. Also the new programming languages and APIs for .NET, including C#, VB.NET and ASP.NET are discussed. Finally attention is paid to database access with ADO.NET.





### .NET Overview





Module 1 : .NET Intro	Module 2 : Managed Execution	Module 3 : ASP.NET
What is .NET? .NET Architecture The .NET Evolution Common Language Runtime What is Managed Code? Common Language Runtime Components .NET Framework Class Library Namespaces ASP.NET and Web Forms Windows Forms and Console Applications XML Web Services ADO.NET: Data and XML .NET Resources	Using a Namespace Defining Namespace and Class Entry Points, Scope, Declarations Console Input and Output Compiling and Running Process of Managed Execution MSIL and Metadata Assemblies Application Domains Garbage Collection	What is ASP.NET? ASP.NET Web Application Web Forms Web Server Controls View State Event Procedures Code Behind Pages Page Event Life Cycle
Module 4 : Web Services	Module 5 : ADO.NET	Module 6 : .NET 4.0
What is a XML Web Service? Why use Web Services? Finding Web Services Creating Web Services Web Service Code Calling Web Service Using HTTP Calling Web Service Using Proxy	What is ADO.NET? Using Namespaces The ADO.NET Object Model What is a DataSet? Accessing Data with ADO.NET DataAdapter Object Model Generating a Dataset Databound Controls	Generics Partial Classes ASP.NET improvements Master Pages

## NET200: Course C# Programming

Code: NET200

Duration: 5 days

### Audience C# Programming Course

The course is designed for experienced developers with a background in Java, C++, Delphi or Visual Basic.

#### **Prerequisites Course C# Programming**

Participants in this course must have experience in Java, C++, Delphi or Visual Basic. Also they are required to know the basics of the .NET Framework.

#### **Realization Training C# Programming**

The theory is presented on the basis of presentation slides. Demos are used to clarify the discussed concepts. The theory is interspersed with exercises. The course material is in English.

### **Certification C# Programming**

Participants receive an official certificate C# Programming after successful completion of the course.

#### **Course C# Programming**

In this course participants learn programming in the .NET platform and the C# language. The emphasis of the course is on the C# syntax, program structure and

implementation details. After attending the course participants will be able to describe the role that C# plays in the .NET Framework and the .NET Platform. They are able to program a simple application, to document it and to compile and debug the application. Participants learn how to use variables, data types, operators, loops, and exception handling. They also learn to call methods and to write methods and to deal with arrays and object oriented programming with classes and objects. In addition the participants learn how to use delegates, events, properties, indexers and attributes. The latest version of C# is used in the course.

Module 1 : C# Intro	Module 2 : Language Syntax	Module 3 : Classes and Objects
What is C#? C# Versions .NET Versions .NET Architecture Common Language Runtime Managed Code C# Compilation Compilation and Execution Managed Execution Assemblies MSIL and Metadata Application Types Garbage Collection .NET Framework Class Library C# Resources	C# Data Types Console IO Variables Variable Scope Case Sensitivity Operators Flow Control if Statement switch Statement for Loops foreach Statement while Statement do Statements break and continue Strings Arrays Methods Parameter Passing	Classes and Objects Example Class and Objects Class Definition Encapsulation Access Modifiers Constructors Creating Objects Fields Properties Special Properties static Modifier Overloading Constants Common Type System Value Types Reference Types Object References
Module 4 : Inheritance	Module 5 : Exception Handling	Module 6 : Namespaces
Inheritance Derived Classes Overriding Methods Hiding Methods Polymorphism Abstract Classes Interfaces Implementing Interfaces Type Casting Implicit Casting Explicit Casting	Error Conditions Exceptions in C# Exception Handling Syntax Exceptions Template Exceptions Object finally Clause Throwing Exceptions User Defined Exceptions Catching User Exceptions	What are Namespaces? .NET Namespaces Defining Namespaces Using Namespaces Nested Namespaces Namespace Aliases Namespace Directory Assemblies Modules MathLibrary Module Assembly Manifest Assembly Manifest AssemblyInfo Using MathLibrary Types of Assemblies Global Assembly Cache Strong Names
Module 7 : Threads	Module 8 : Synchronization	Module 9 : Special Classes





### C# Programming

Multiple Threads Benefits and Drawbacks Thread Characteristics C# Thread Model Thread Class Thread Stack Thread Delegate Multiple Threads Autonomous Classes Passing Parameters Thread Naming Background Threads Thread Exceptions Thread Methods	Concurrent Method Invocation Synchronization Blocking on Monitor Lock Statement Mutual Exclusion in C# Joining Threads Interrupting Threads DeadLock Wait Handles Interthread Communication Condition Synchronization Monitor Wait and Pulse	What is a Delegate? Benefits of Delegates Multicasting Delegates and Events Simple Event Handling Enumerations Enumeration Base Types Extension Methods Partial Classes Attributes Attribute Parameters Custom Attributes Nullable Types Static Classes
Module 10 : Utility Classes	Module 11 : Generics	Module 12 : Collections
Object Class Boxing and Unboxing Overriding Equals Math Class DateTime Structure Regex Class Input Conversion Convert Class Process Class Environment Class Globalization Localizing Dates Localizing Numbers XML Documentation	What are Generics? Need for Generics Generic Class Syntax Benefits of Generics Multiple Generic Parameters Runtime Type Parameter Constraints Generic Methods	What are Collections? Framework Classes Properties of Collections Predefined Collections Array Class List Class Queue Class Queue Methods Stack Class Stack Methods Linked List Sorted List Dictionary Hashtable BitArray

### Module 13 : File I/O

Stream I/O I/O Classes File Types Writing Text File Reading Text File Using Directive Accessing Binary Files Buffered Streams Serialization Implementing Serialization Accessing File System Directory Classes

## NET300: Course ASP.NET Web Development

Code: NET300

Duration: 4 days

### Audience ASP.NET Web Development Course

This course is intended for Web developers who want to learn ASP.NET 4 and Visual Studio.

### Prerequisites Course ASP.NET Web Development

To participate in this course a good knowledge of HTML and basic knowledge of C# is required.

#### **Realization Training ASP.NET Web Development**

The course has a hands-on nature. The theory is treated on the basis of presentation slides. The theory is interspersed with demos and exercises. The course materials are in English.

### **Course ASP.NET Web Development**

In the course ASP.NET Web Development participants will get a comprehensive introduction to ASP.NET with which they are able to develop modern state of the art web applications. After completing the course participants will understand the important aspects of ASP.NET and apply these in their own web applications. Attention is paid to building a User Interface in ASP.NET that uses Master Pages and all sorts of controls. Next the possibilities ASP.NET has for input validation of these controls are discussed. Accessing databases with ADO.NET from ASP.NET



Pnet

ASP.NET Web Development



applications is on the program as well. An important aspect of Web applications is how user data can be maintained between requests and how user sessions can be made. This is discussed in the section on State Management. Finally attention is paid to the Entity Framework which implements Object Relational Mapping.

Module 1 : Intro to ASP.NET	Module 2 : ASP.NET UI	Module 3 : Validation Controls
Overview of ASP.NET HTML Controls Web From Server Controls How Web Forms Work IIS Objects Creating User Controls Constructing Web Forms with controls Developing with Visual Studio Debugging Applying the common Web Form controls	Ensuring uniform page layout Devising site templates Adding controls to Master pages Implementing ASP.NET Menu, TreeView and SiteMapPath Defining your site structure using SiteMaps Dynamically moving between pages in code Implementing CSS without changing page content Building Themes from Skins and CSS Styling controls with Skins	Introducing Validation Controls Using the RequuredFieldValidator Control Using the RangeValidator Control Using the RegularExpressionValidator Control Using the CustomValidator Control Using the ValidationSummary Control Using the CompareValidatorControl
Module 4 : ADO.NET	Module 5 : State Management in ASP.NET	Module 6 : Entity Framework
Getting Started with ADO.NET Benefits of ADO.NET Introducing the DataReader Using a Data Reader to Fill a DropDownList Control Retrieving DataSets Generically Working with Relations in a Database Introducing ADO.NET Connections Providing ADO.NET Connection Information Using ADO.NET Connection Objects Updating Data Using a Command Object	Investigating State Management Using the Session Object Using Cookies ViewState Property Cookieless Sessions ASP.NET State Service SQL Server State Management What's Changed from ASP	Reporting Generating selection criteria Filtering data Reporting from dynamically cached data Binding the Entity Framework Integrating data Entity DataSource Dynamically generating queries Query Extender Selecting and sorting data with LINQ



## NET400: Course Web Services in .NET

Code: NET400

#### Duration: 4 days

### Audience Web Services in .NET Course

This course is intended for developers who want to understand and use .NET WCF Web Services in their applications.

#### Prerequisites Course Web Services in .NET

To participate in this course knowledge and experience with C# is required and knowledge of ASP.NET is beneficial for a proper understanding.

#### **Realization Training Web Services in .NET**

The course has a hands-on nature. The theory is treated on the basis of presentation slides. The theory is interspersed with demos and exercises. The course materials are in English.

### **Certification Web Services in .NET**

Participants receive an official certificate Web Services in .NET after successful completion of the course.

#### **Course Web Services in .NET**

In the course Web Services in .NET participants learn to develop Web services using Windows Communications Foundation (WCF) and C#. In the course the fundamentals

of the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL), essential for creating interoperable Web services, are discussed. The features and function of Web services are detailed. This includes the tracing of a simple Web service, examining HTTP, XML, SOAP and WSDL. The course covers the techniques for creating and debugging ASP.NET Web services using Visual Studio .NET, and creating clients for Web services using the .NET API directly. More advanced topics, such as state management, caching and transactions in Web services are are treated. The key technologies of ASP.NET Web services including XML serialization, SOAP, WSDL and Universal Description, Discovery and Integration (UDDI) are on the course program. The .NET classes for manipulating WSDL files are also examined. Next Web services security and new emerging Web Services specifications are explored. This includes the various security issues and technologies in Web services us an HTTP, HTTPS, SSL 3.0, XML Signature, XML Encryption, XML Key Management Specification (XKMS), Security Assertion Markup Language (SAML) and WS-Security.





### Web Services in .NET





Module 1 : WCF Intro	Module 2 : Web Services Intro	Module 3 : SOAP
What is WCF? Positioning WCF WCF versus Web Services Endpoints Addresses WS-Addressing Addressing Examples Bindings WCF Bindings Configuring Bindings WCF Contracts Service Contracts Data Contracts Data Contracts Fault Contracts Fault Contracts Creating Endpoints Multiple Endpoints Hosting WCF Services WCF Architecture	What are Web Services? Distributed Applications Evolution Role of interface RPC Example Interoperability Web Service Types Web Services Stack SOAP Web Services REST Web Services REST Web Services Document Style Web Services Service Oriented Architecture	What is SOAP? SOAP Characteristics SOAP Design Goals SOAP Protocol Concepts SOAP Messages SOAP Body SOAP Headers SOAP Namespaces SOAP Faults SOAP Faults SOAP Version differences SOAP Message as payload Message Exchange Patterns SOAP Message Path SOAP Message Path SOAP Intermediaries actor and mustUnderstand attribute
Module 4 : XML-Schema	Module 5 : WSDL	Module 6 : Hosting
Why XML-Schema? Well formed and valid documents What XML-Schema's? Markup Languages XML Schema Advantages XML Schema design models Classic Use of Schema's XML Namespaces Simple and Complex types XML Schema Data Types User Defined Data Types Derivation by Restriction Derivation by Extension	What is WSDL? Where is WSDL used? Benefits of WSDL WSDL and Code Generation WSDL in Web Service stack WSDL Namespaces WSDL Structure WSDL Elements Types and Messages PortType and Operations WSDL Bindings Service Element SOAP Messages Modes WSDL 2.0	What is Hosting? Hosting Types Hosting Criteria Service Description Self Hosting Service Host Creation App.config Configuration Programmatic Configuration Windows Host Service Description Managed Window Service Creating Windows Services Hosting in Windows Services IIS Hosting .SVC File Web.config for IIS Host Hosting in IIS Windows Activation Service WAS Commands
Module 7 : Contracts/strong>	Module 8 : Instance Management	Module 9 : Binding

WCF Contracts Service Contract Creating Service Contract Data Contract Employee Data Contract Service Implementation Client Side Message Message Pattern Message Contract Definition Message Contract Rules Customizing SOAP MessageHeaderArray ProtectionLevel Property Name Property Order Property Fault Contract	Instance Management Instance Mode Configuration Per Call Service Process of Handling Per Call Per Session Service Singleton Service Instance Deactivation ReleaseInstanceMode ReleaseInstanceMode.None BeforeCall BeforeAndAfterCall Explicit Deactivation Durable Services Defining Durable Services Throttling Throttling Configuration Programmatic Configuration	Binding Bindings and Channel Stacks Message Bubbling Basic Binding Types WS Binding Types NET Binding Types Binding Configuration Administrative Configuration Programmatic Configuration Metadata Exchange Publishing Metadata Metadata Exchange Point MEX Administrative Config MEX Programmatic Config
Module 10 : Message Patterns	Module 11 : Web API REST Services	
Message Patterns Request-Reply One Way One Way Operation Sessionful Services Exceptions Callback Service Callback Contract Client Callback Setup Service Side Callback Invocation	What is REST? REST Web Services Simple REST Examples REST Web Service Principles ID and Links REST Services with Web API Multiple Representations Embedded Path Parameters Common REST Patterns Resources URI Access JavaScript Object Notation (JSON) XML versus JSON	

## **NET500: Course XAML Programming**

Code: NET500

#### Duration: 2 days

### Audience XAML Programming Course

This course is designed for experienced .NET developers who want to use XAML to develop .NET user interfaces.

### Prerequisites Course XAML Programming

To join this course knowledge and experience with .NET application development is required.

### **Realization Training XAML Programming**

The theory is discussed on the basis of the presentation slides and is interspersed with practical exercises. Demos are used to illustrate the concepts. The course material is in English.

### **Course XAML Programming**

The course XAML Programming covers how XAML user interfaces for .NET applications can be built using Extensible Application Markup Language or XAML. In particular, attention is paid to the use of XAML in a Windows Presentation Foundation (WPF) environment. After an introduction about where XAML is used and why XAML exists, it is discussed how you define visual elements in declarative XAML markup. XAML markup is thereby separated from the business logic in code behind files which contain partial class definitions. The syntax of the XAML language and the various





### XAML Programming





objects and their properties are also discussed. Attention is further paid to data binding and how events in user interface elements can be attached to code. The knowledge of XAML obtained in this course can also be applied elsewhere.

Module 1 : XAML Intro	Module 2 : XAML Basics	Module 3 : XAML Graphic Elements
What is XAML? WPF Architecture Drawing with XAML Where is XAML used? XAML Properties Markup extensions Why XAML? Graphics and Imaging Audio and Video Support XAML on the Web XAML tools XAML alternatives	XAML versus Code Attributes and Events Nesting Elements XAML Namespaces WPF Properties Type Converters Property Mini Language Markup Extensions Nesting Controls Content Collections Naming Elements Adding Events with names	Basics Graphics Element Canvas Parent Positioning Shapes Brushes Brush Types Using Text Images Transformation Transformation Types Combining Transformations Media Integration
Module 4 : Animation	Module 5 : 3D	Module 6 : Data Binding
Animating with XAML StoryBoards Animation Example Triggers DoubleAnimation ColorAnimation PointAnimation Animation with Keyframes Types of Keyframes Programmatic Animation	WPF 3D Viewport 3D Viewport 3D contents XAML Properties Camera Type Camera Point of View Light Model GeoMetryModel3D Materials : 3DBrushes Transformations 3D and Feasibility	Need for Data Binding Data Binding Singular Binding Simple Binding Conversions Validation Data Template Master Detail Filtering Sorting

## NET600: Course Ajax .NET Programming

Code: NET600

#### Duration: 2 days

### Audience Ajax .NET Programming Course

This course is intended for experienced .NET developers who want to use Ajax to develop Web applications.

#### Prerequisites Course Ajax .NET Programming

To join this course knowledge of and experience in .NET Web Application Development with ASP.NET is required.

### **Realization Training Ajax .NET Programming**

The theory is discussed on the basis of presentation slides and is interspersed with practical exercises. Demos are used to clarify the concepts. The course material is in English.

### **Course Ajax .NET Programming**

The course .NET Ajax programming focuses on the use of Ajax in an ASP.NET environment. After an introduction to the basics of Ajax and the XMLHttpRequest object and an overview of the technologies that Ajax combines such as JavaScript, DOM and Dynamic HTML, the Microsoft Ajax Framework is discussed in ASP.NET. In this respect the role of the Script Manager will be discussed and also the Microsoft Ajax extensions. Attention is also paid to the various controls that the Microsoft Ajax library offers like the UpdatePanel and the Timer. Finally JavaScript Object Notation





### Ajax .NET Programming



ASPINET AJAX Futures

(JSON) for the transfer of JavaScript objects over the network and the invocation of Web Services from Javascript in ASP.NET pages will be discussed.

Module 1 : AJAX introduction	Module 2 : AJAX API Details	Module 3 : ASP.NET Ajax
What is AJAX? AJAX Technologies Basic AJAX API XHTML Embedding XHTML XML ??Documents JavaScript DOM API DOM hierarchy	Request Object GET and POST Requests Concurrent Requests Event Handling Dispatching an Event Serializing DOM Document as Text Posting an XML Document Processing XML From an ASP page	Architecture of ASP.NET Ajax Partial Rendering Ajax Library to Enhance User Interfaces Role of Script Manager Types of System Extensions ASP.NET Ajax Extensions Script Manager Proxy UpdatePanel Progress Update Timer Ajax Toolkit
Module 4 : JSON	Module 5 : Calling Web Services	
JavaScript Object Notation (JSON) JSON Syntax JSON Based Communication JSON: Server Side JSON: Client Side Working With Arrays JSONObject Methods	JavaScript proxies Web.config entries Service Reference Bridge files Provide Bridge Build	

## **PHP100: Course PHP Programming Fundamentals**

Code: PHP100

Duration: 3 days

### Audience PHP Programming Fundamentals Course

The course **PHP** Programming Fundamentals is intended for Web developers who want to learn how to use PHP (PHP Hypertext Preprocessor) for writing dynamic Web pages.

#### **Prerequisites Course PHP Programming Fundamentals**

To participate in this course <u>HTML</u> knowledge is required. Programming knowledge is beneficial for a rapid understanding.

### **Realization Training PHP Programming Fundamentals**

The concepts are discussed on the basis of presentation slides and demos. The theory is interspersed with exercises. The course is done with PHP 7 which is installed in a WAMP package with the Apache Web server and MySQL.

### **Certificate PHP Programming**

Participants receive an official certificate PHP Programming after successful completion of the course.

### **Course PHP Programming Fundamentals**

The course PHP Programming Fundamentals covers the basic principles of the

HTML-embedded scripting language PHP. PHP stands for Personal Hypertext Preprocessor, is available on Windows and Linux and is intended for developing dynamic Web pages. The course discusses PHP version 7 and also the differences between PHP 5 and PHP 7 are discussed. PHP borrows much of its syntax from C, Java and Perl and also has an extensive library of functions. During the course participants work on a case study which consists of a number of follow-up exercises. Participants learn to program with variables, data types, operators and control flow constructs of the PHP language. PHP functions and arrays are also addressed. Central element of the course is the way to generate dynamic Web pages with PHP. Attention is also paid to submitting Web Forms, reading input parameters, handling errors, handling cookies and working with sessions. Finally it is discussed how PHP can be used for accessing MySQL databases and files on the server. An optional module, if time permits, is the validation of input forms using PHP. In conjunction with the course <u>Advanced PHP Programming</u> this course covers the requirements for the <u>Zend PHP Certification</u> exam.

Module 1 : PHP Introduction	Module 2 : Variables and Operators	Module 3 : Control Flow
What is PHP? History of PHP PHP Usage Statistics Performance Comparison PHP Benefits PHP Processing Model The Script Tags PHP Basics PHP Output Statements PHP Installation PHP Configuration PHP IDE's	PHP Constants PHP Variables PHP Data Types Weak and Dynamic Typing Number Data Type String Data Types Boolean Data Types PHP Operators Variable Interpolation Include and Require Indexed Arrays Associative Arrays	if Statement else Statement switch Statement for Loop while Loop do while Loop break Statement continue Statement Nested Loops foreach Loops
Module 4 : Functions	Module 5 : Submitting Forms	Module 6 : Error Handling
User Defined Functions Variable Scope Static Variables Case Sensitivity References Call by Reference Call by Value Terminating scripts Number Formatting PHP Library Functions Numerical Functions String Functions Array Functions	HTTP Requests and Responses HTML Forms Form Submission Using POST and GET PHP Form Processing Using isset Submit on Self HTML Input Types Accessing Form Parameters Multiple Valued Form Elements HTTP Request Headers HTTP Status Codes PHP Superglobals	Debugging Syntax Errors Logic Errors Displaying Errors Error Severity Levels Error Configuration Settings Error Reporting Logging Errors User Input Syntactical Errors Semantical Errors Fatal Errors Non Fatal Errors Error Handling
Module 7 : Cookies and Sessions	Module 8 : File I/O	Module 9 : Database Access
What are Cookies? Problems with Cookies Getting Cookies Deleting Cookies Cookie parameters Setting Cookie parameters Why Session Tracking? Session Tracking Session Tracking Session Tracking Mechanisms PHP Sessions	PHP Files Overview Creating and Opening Files Reading Files Writing Files Deleting Files Navigate within a File fgets Function file_get_contents Function file_put_contents Function Uploading Files Handling Uploaded Files Uploaded File Location	PHP and Data Access What is mysqli? Connecting to MySQL mysqli Connection Functions mysqli Query Functions Data Definition Query Inserting Records Retrieving Results Fetch Functions Buffered Queries Unbuffered Queries Prepared Statements
Optional Module : Validation		



php

PHP Programming Fundamentals



Hidden Field to Enable Validation Validating Form Data Displaying Error Messages Required Fields Integer Fields Checking an Integer Range Floating Point Fields Checking an Email Address Default Values Checking Login Data

## PHP303: Course Advanced PHP Programming

Code: PHP303

Duration: 4 days

### Audience Advanced PHP Programming Course

The course Advanced PHP Programming is intended for <u>PHP</u> web developers with basic knowledge of PHP who want to make advanced use of PHP to write dynamic Web applications.

#### Prerequisites Course Advanced PHP Programming

To participate in this course knowledge and experience with PHP is required as is discussed in the course **PHP Programming**.

### Realization Training Advanced PHP Programming

The concepts are treated on the basis of presentation slides. Demos are used to clarify the theory. The theory is interspersed with exercises. The course uses PHP 7 which is installed in a WAMP package with the Apache Web server and MySQL.

### **Certification Advanced PHP Programming**

Participants receive an official certificate Advanced PHP Programming after successful completion of the course.

#### **Course Advanced PHP Programming**

The course Advanced PHP Programming covers advanced features of PHP. In the

first place it is addressed how to deal with Object Orientation in PHP. Attention is paid to the known object oriented concepts like classes and objects, encapsulation and inheritance and how PHP deals with them. Also the concept of Exception Handling is covered. The new features of PHP 7 are discussed as well. Another issue is how to handle XML and JSON in PHP. Various XML libraries in PHP are discussed, such as libraries for SAX, DOM, SimpleXML and XPath. Also attention is paid to the handling of JSON data with the JSON library functions. The use of regular expressions in PHP is also part of the subject matter. Attention is paid to various libraries for database access such as PDO, PHP Data Objects. Database abstraction, transactions and SQL joins are also discussed. Further attention is payed to several aspects of security such as security threats, authentication and SSL. In the following module it is discussed how interactive PHP Ajax Web Applications can be developed. The XAJAX Framework is addressed in this respect. In the module Web Services both SOAP and REST Services are discussed. Finally an overview of Design Patterns is given and a number of Design Patterns in PHP applications are treated. In conjunction with the course **PHP Programming** this course covers the requirements for the **Zend PHP Certification** exam.

Module 1 : Classes and Objects	Module 2 : Inheritance	Module 3 : Exception Handling
Object Oriented Programming Class Definition Creating Objects Encapsulation and Data Hiding Private Data Constructors and Destructors \$this Variable Default Field Values Static Members Class Constants Object References Cloning Objects	Derived Classes Base Class Access Constructor Visibility Overriding Methods Pseudo Constants Final Classes and Methods Abstract Classes Implementing Interfaces instanceOf Operator Polymorphism Type Hinting Catchable Type Errors	Types of Errors Exceptions in PHP try, catch and finally Exception Handling Rules Exception Class Exception Specialization User Defined Exceptions Multiple catch Clauses Standard PHP Exceptions Main Exception Branches RunTime Exceptions Turn Errors into Exceptions
Module 4 : PHP 7 Features	Module 5 : XML and JSON	Module 6 : Regular Expressions
Scalar Type Declarations Coercive and Strict Mode Return Type Declarations Null Coalescing Operator Spaceship Operator Constant Arrays Anonymous Classes Closure::call() Filtered unserialize() Expectations Use Statement Error Handling	XML Extensions for PHP SimpleXML Load XML from File and String Reading Attributes and Elements Creating XML Data Document Object Model DOM Validation SAX, Simple API for XML Using XPath in PHP PHP JSON Functions JSON Syntax Encoding and Decoding JSON	Regular Expressions in PHP PHP Regular Expression Functions Meta Characters Quantifiers and Character Classes Regular Expression Modifiers Matching Text with preg_match Matching on Word Boundaries Matching Text with preg_match_all Greedy and Non-Greedy Quantifier Replacing Text with preg_replace Splitting Text with preg_split Searching Text with preg_grep
Module 7 : Databases Access	Module 8 : Security	Module 9 : PHP and Ajax
PHP Data Objects Data Access with PDO Creating and Closing Connections PDO Error Retrieval Select Queries Fetching Results Direct Query Execution Array and Column Fetching PDO Transactions Prepared Statements Calling Stored Procedures	Security Issues Register Globals Spoofed Form Submissions Cross Site Scripting Validating Input SQL Injection Cross Site Request Forgery HTTP Authentication SSL Overview SSL Handshakes Configuring SSL in Apache	Classic Web Application Model Ajax Web Application Model Typical Ajax Interactions Drawbacks of Ajax Creating the XMLHttpRequest Object Methods Sending the Request Object Properties XMLHttpRequest readyState Handling the Response PHP and Ajax, XAJAX
Module 10 · Web Services	Module 11 : Design Patterns	







REST versus SOAP SOAP Messages WSDL and Code Generation PHP SOAP Extension WSDL for SoapService PHP SOAP Services REST Services Everything an Id Standard HTTP Methods Content Negotiation What are Patterns? Singleton Pattern Defined Singleton Pattern Usage Adapter Pattern Factory Patterns Factory Method Pattern Abstract Factory Pattern Abstract Factory Usage Observer Pattern Iterator Pattern

## PHP401: Course PHP Programmer Certification

Code: PHP401

Duration: 2 days

### Audience PHP Programmer Certification Course

The course PHP Programmer Certification is intended for experienced PHP developers that want to prepare themselves for the PHP 7 Programmers exam.

#### Prerequisites Course PHP Programmer Certification

Knowledge and experience with Web development and <u>PHP programming</u> and <u>Advanced PHP Programming</u> is required to participate in this course.

#### **Realization Training PHP Programmer Certification**

The course has a hands-on nature. The theory is interspersed with practical exercises. Several mock exams with exam test questions are discussed. The course material is in English.

### **Certification PHP**

Participants receive an official certificate PHP Programmer Certification after successful completion of the course.

#### **Course PHP Programmer Certification**

In the course PHP Programmer Certification the topics for the PHP 7 Programmers

exam are discussed on the basis of test questions. The course also has the purpose

to give the best possible impression of what one can expect on the actual exam. The course is an exam-oriented training and programming assignments are not part of the course program. Participants can however experiment with demo scripts where certain aspects of the exam are discussed. Extra attention is paid to topics that are commonly experienced as difficult. The exam topics on the program are: PHP Basics, Object Oriented Programming, new features in PHP 7, stream I/O and network programming, using functions and arrays, working with strings and regular expressions, accessing databases and SQL, accessing XML data and Web services, security and finally the use of Design Patterns.

Exam Topic: PHP Basics	Exam Topic: Date Formats and Types	Exam Topic: PHP 7 Features
Arithmetic and Bitwise Operators Assignment Operators Comparison Operators String and Array Operators Logical Operators Variables Condition Control Structures Loop Control Structures Special Constructs Constants Namespaces Extensions Configuration and Performance	XML Basics XML Extension Character Encodings SimpleXML SimpleXML Methods DOM SOAP and REST SOAP Constants REST Principles REST Context Switching JSON JSON Constants DateTime Methods	Scalar Type Declarations Coercive and Strict Mode Return Type Declarations Null Coalescing Operator Spaceship Operator Constant Arrays Anonymous Classes Closure::call() Filtered unserialize() Expectations Integer Division Use Statement Error Handling
Exam Topic: Strings	Exam Topic: Arrays	Exam Topic: Input and Output
Strings Comparing Strings String Functions Formatting Output Regular Expressions RegEx Syntax Encoding	Arrays Array Operations Looping Arrays Array Methods Sorting Arrays Other Array Functions ArrayObject Class	Files File System Functions File Operations Streams File Wrappers Stream Contexts Reading and Writing
Exam Topic: Object Orientation	Exam Topic: Databases	Exam Topic: Security
Classes and Objects Inheritance Abstract Classes and Interfaces Exceptions Properties and Methods Static Modifier Autoloading and Reflection Type Hinting Late Static Binding Magic Methods SPL Generators and Traits	Databases SQL Statements Queries SQL Functions Joins Prepared Statements Transactions PDO PDO Connections PDO Queries PDO Queries PDO Statements Parameter Passing	Configuration Built-in Safeguards PHP as Apache Module Session Security Cross-Site Scripting Cross-Site Request Forgeries SQL Injection Remote Code Injection Dynamic Data Attacks Input Filtering and Escaping Output Password Hashing API SSL
Exam Topic: Functions	Exam Topic: Web Features	Exam Topic: Error Handling



**PHP Programmer** 



Function Definition Declaring Functions Function Arguments Variable Scope Variable Functions Anonymous Functions Closures Type Declarations Sessions Forms Encoding and Decoding FILE Uploads Cookies Cookie Handling HTTP Headers and Codes HTTP Authentication Error Levels Error Display Error Configuration Error Logging User Defined Errors Exception Handling Exception Hierarchy Error Class

## PHP500: Course PHP Zend Framework Programming

Code: PHP500

Duration: 4 days

### Audience PHP Zend 2 Framework Programming Course

Experienced PHP developers who want to learn the capabilities of the Zend Framework.

#### Prerequisites Course PHP Zend 2 Framework Programming

Extensive knowledge and experience with PHP development including object oriented programming is required.

### **Realization Training PHP Zend 2 Framework Programming**

The concepts are treated on the basis of presentation slides and demos. The theory is interspersed with exercises. A modern PHP IDE is used for the demos and exercises. The focus is on Zend Framework version 2.

### **Course PHP Zend 2 Framework Programming**

The Zend 2 Framework is an open source PHP class library focusing on simplifying complex PHP applications through the use of standard components. The course starts with an overview of the development of PHP applications based on the Zend 2 Framework and discusses the architecture of the Zend Framework. The focus is on Zend Framework version 2. The next subject is the Model View Controller (MVC) design pattern that is a central element for PHP applications based on the Zend Framework. Attention is paid to the function of the Controller component for the routing

of requests and the control flow. This will also include input validation, authentication, authorization and security best practices. Furthermore, the function of the Model component is discussed, with attention being paid to accessing databases, caching of data and sending mail. Also the function of the View component is discussed and this includes the addressing of Rich User Interfaces and Ajax functionality. Finally the subjects internationalization and Web Services are on the

discussed and this includes the addressing of Rich User Interfaces and Ajax functionality. Finally the subjects internationalization and Web Services are on the course schedule. The topics covered in the course are also the exam subjects for the Zend Framework Certification exam. The course is not a direct exam preparation training, but a training to learn how to use the Zend Framework in PHP Development.

Module 1 : ZF Intro	Module 2 : ZF MVC	Module 3 : Event Managers
What is Zend Framework? ZF2 Key Features ZF2 Architecture ZF2 Architecture and MVC Zend-Tool Decoupling Namespaces Autoloading Project Structure Deployment in Virtual Host Front Controller Rewrite Module Zend Framework Control Flow	What is MVC? Design Patterns MVC Elements and Implementation ZF2 MVC: Event-Driven Architecture Key MVC Events Services in MVC Zend_Controller_Front Request Routing Action Controllers Modules Zend_View Rendered View Error Controller	Events Event Types Event Manager Trigger Events Available Methods Aggregates Shared Event Listeners Identifiers Shared Event Manager ModuleManager Events
Module 4 : MVC and Models	Module 5 : Controllers	Module 6 : Routing
Application Structure Modules in MVC Structure of a Module Bootstrapping Modular Application MVC-related Module Configuration ZendModuleManager ModuleManager Listeners Autoload Files for Modules Module Classes MVC 'bootstrap' Event ModuleAutoloader DefaultListenerAggregate Module Best Practices	Zend_Controller Components Request Object Accessing a Request and Response Request Handing Workflow Request Object Operations Request Type Checkers Zend_Controller_Front Dispatch Loop Front Controller Parameters MVC Events and Controllers MVC-based Action Controllers AbstractActionControllers Interfaces Controller Plugins Registering Module-specific Listeners	Routing Basics ZF2 Router Types ZF2 HTTP Route Types Base Url Standard Router Using Rewrite Router Shipped Routes Dispatcher Using _forward Action Helpers Plugins View Scripts
Module 7 : View Layer	Module 8 : Forms and Validation	Module 9 : Database Access



RAMEWORK

## PHP Zend Framework Programming



ZEND

FRAMEWORK

Zend View View Model Layouts Alternate Rendering Response Strategies What are Layouts? Typical Layout Two Step View Pattern Simple Layout template Zend_Layout Changing Layouts Partials Partials Parameters Partials and PartialLoop Place Holders PartialLoop Model PHP Renderer View Helpers	Zend InputFilter Zend Input Filtering Zend Form Validating Forms Rendering Forms Creating Forms Zend_Form_Elements Base Classes Plugins Utilities Zend Validator Validators Decorators Decorating Process Sub Forms	Connecting to a database Zend_DB Query Preparation Query Execution Running an SQL Query Zend Db Adapter Creating an Adapter Creating Statements Zend Db Sql Zend Db TableGateway Fetching Results Changing the Fetch Mode CRUD Operations Quoting Retrieving MetaData Prepared Statements Zend_DB_Select Table and Row OO patterns
Module 10 : ZF Services	Module 11 : ZF Security	Module 12 : ZF Components
Service Manager (SM) Service Types Configuring Services What is a Web Service? Kinds of Web Services ZF Soap Services Zend_Soap Server Zend_Soap Client XML-RPC XML-RPC in ZF JSON-RPC JSON-RPC in ZF REST REST n ZF Zend_Rest Server	Security Concerns Authenticating with Zend_Auth Zend_Auth Adapters Authenticating Results Persisting Identities Customized Storage Database Table Authentication Zend_Auth Adapter_DbTable Basic and Digest Authentication HTTP Authentication Adapter LDAP Adapter Zend_ACL Access Control Lists Roles and Resources	Components Zend Framework ZF Component Dependencies Zend Core and Components Why ZF Components Zend_Paginator Zend_Config_Ini Zend_Registry Zend_Pdf Zend Mail Zend_Log Zend_Locale

## PHP550: Course Laravel Framework Programming

Code: PHP550

Duration: 4 days

### Audience Laravel Framework Programming Course

PHP developers who want to develop robust and maintainable Web Applications with the Laravel PHP Framework.

#### Prerequisites Course Laravel Framework Programming

Experience with PHP Programming and Web Development with PHP is required to participate in this course.

### **Realization Training Laravel Framework Programming**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

### **Certification Laravel Framework Programming**

The participants receive an official certificate Laravel Framework Programming after succesfull completion of the course.

#### **Course Laravel Framework Programming**

In this course participants learn how to build modern PHP applications with the state of the art Laravel Framework. Laravel has evolved in recent years into one of the most

popular PHP frameworks. This is due to the ease of use, the expressive syntax, the dependency injection mechanism and directly deployable components. In this course you will learn to install Laravel and prepare an environment for application development. Laravel's routing architecture is discussed, the handling of requests and the generation of responses. Also Laravels Model View Controller architecture as well as Forms and validation are treated next. The attention then goes to database access in Laravel and Object Relational Mapping with Laravel's Eloquent ORM. Finally advanced topics such as authentication and security are discussed. At the end of the course the participants have gone through various case studies with Laravel and they are able to create robust Web sites with Laravel.

Module 1 : Laravel Intro	Module 2 : Routing	Module 3 : Controllers and Filters
What is Laravel? Laravel Installation Creating a Laravel project Laravel Structure Overview Project Structure Facades MVC Pattern Dependency Injection Routing Annotations Lavarel Versions Introducing Composer Using Homestead	Laravel Request Lifecycle Basic Routing Route Parameters Restricting Parameters Handling HTTP Exceptions Returning Responses Views View Data Redirects Custom Responses Resource Controllers Blade Templates	Creating Controllers Controller Routing Controller to Command Command to Event Queued Event Handlers Nested Controllers Route Caching RESTful Controllers Basic Filters Multiple Filters Filter Classes Global Filters Pattern Filters
Module 4 : Forms and Validation	Module 5 : Database Access	Module 6 : Eloquent ORM
Master Template Forms and form tag Text and labels Buttons Closing the Form Security Validation Rules Error Messages Custom Rules Custom Message Generating Framework URLS Generation Shortcuts	Abstraction Configuration Preparing Creating Tables Column Types Special Column Types Column Modifiers Updating Tables Dropping Tables Schema Tricks Basic Concept Creating Migration Rolling Back Migration Tricks	Creating new model Reading Existing Models Updating Existing Models Deleting Existing Models Queries Preparation Eloquent to string Query Structure Magic Queries Query Scopes Relationships Implementing Relationship Relating and Querying
Module 7 : Rest API	Module 8 : Authentication and Security	
Rest Services Resource ID's Create and Update Read and Delete List Pagination Model Binding Nested Controllers Beyond CRUD Nested Update Nested Create	Authenticating users User model Authenticating routes Authenticating views Validating user Input Avoiding mass assigment Cross Site Scripting Cross Site Request Forgery SQL Injection Forcing HTTPS	





## PHP580: Course Symfony Framework Programming

Code: PHP580

Duration: 4 days

(\$

X

Symfon

### Audience Symfony Framework Programming Course

PHP developers who want to develop robust and maintainable Web Applications with the Symfony PHP Framework.

#### Prerequisites Course Symfony Framework Programming

Experience with PHP Programming and object oriented development with PHP is required to participate in this course.

#### **Realization Training Symfony Framework Programming**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

### **Certification Symfony Framework Programming**

The participants receive an official certificate Symfony Framework Programming after succesfull completion of the course.

### **Course Symfony Framework Programming**

In this course participants learn how to build modern PHP applications with the state of the art Symfony Framework. Symfony has evolved in recent years into one of the

most popular PHP frameworks. This is due to the ease of use, the good performance, the dependency injection mechanism and directly deployable components. In this course you will learn to install Symfony and prepare an environment for application development. Symfony's routing architecture is discussed, the handling of requests and the generation of responses. Attention is paid to Twig templates en template inheritance. Next the Doctrine Object Relational Mapping implementation is discussed. Using Doctrine PHP classes and their relations can be stored in the database. Symfony offers many facilities for Forms and these are discussed extensively. Also the validation of forms is treated. Attention is paid to service classes, the service architecture and the registration of services with the container. The security facilities that Symfony has to offer, like authentication and authorization with the User Entity class are discussed as well. Also part of the subject matter are advanced subjects like hooking into the routing mechanism and the implementation of dependency injection. Finally attention is paid to the creation of a Rest API with Symfony. At the end of the course you will have completed an extensive case study with Symfony and you are able to create robust Web applications with Symfony.

Module 1 : Symfony 3 Intro	Module 2 : Bundles	Module 3 : Data Access
Why Symfony? Installing Symfony Console commands Namespaces Controller and Route How bright is the future? Installation Composer and Packagist Installing Symfony via Composer The road map Checking the installation	Request and Response Life Cycle Anatomy of a bundle Generating a new bundle Best practices Custom bundles versus AppBundle Creating templates with TWIG Controller/View interaction Conditional templates Make it dynamic Installing bundles Creating data fixtures Loading data fixtures Defining and prioritizing features	Database configurations Generating an entity What is Object Relational Mapping Entity Classes Using doctrine Inserting new Objects Adding Columns Updating Table Schema Database Migrations Migrations Workflow Nullable Columns
Module 4 : Caching in Symfony	Module 5 : Presentation Layer	Module 6 : Security in Symfony
Caching in Symfony Definition of a cache Characteristics of a good cache Caches in a Symfony project Key players in the Reverse proxy cache Set expiration for dashboard page Validation strategy Expiration stategies Validation strategies Doctrine cache ESI for selective caching Sophisticated bundles	Asset management How templates are organize Navigate or not to navigate What is Bootstrap? MopaBootstrapBundle Bootstrap configuration Creating menus Rendering the menu Dashboard template Overriding templates Profile-related templates Changing the backend logo	Authentication Authorization User Class Dashboard and Security Security is organization Authentication Authorization FOSUserBundle Security settings Adding FOSUserBundle Adding routes
Module 7 : Testing	Module 8 : Forms and Validation	Module 9 : Rest API
TDD and BDD with Codeception Creating a functional test Developing the missing code Creating the unit tests Setting up the database Recreating up the database for test Creating unit tests Writing code to pass test Running functional and unit tests On the CI side of the story	Form Input Filtering Validating Forms Rendering Forms Creating Forms Save and Redirect setFlash and Dance Flash Messages Utilities Validators Decorators Decorators Decorating Process Sub Forms	What is REST? Rest Services Resource ID's REST Web Services Simple REST Examples REST Web Service Principles Multiple Representations Embedded Path Parameters Common REST Patterns Resources URI Access JavaScript Object Notation (JSON)

## **PHP Symphony Framework**





X

## PHP600: Course WordPress Web Development

Code: PHP600

Duration: 2 days

### Audience WordPress Web Development Course

This course targets persons that want to use WordPress for the design of a Web site or Web Application.

#### Prerequisites Course WordPress Web Development

No specific knowledge is required to participate in this course. General knowledge of Web applications and experience with computers is desired.

#### **Realization Training WordPress Web Development**

The theory is discussed on the basis of presentation slides. Demo's are used to clarify the treated concepts. In a number of subsequent exercises participants create a WordPress Web site with often used functionality.

### **Course WordPress Web Development**

In the course WordPress Web Development participants learn how the open source Content Management System WordPress can be used to develop different types of Web Sites. The course starts by downloading and installing WordPress from scratch and creating a Home page. Next the administration area called the Dashboard of the site is explored. The different content types that can be placed in WordPress are discussed as well. The participants will learn about pages, blog items, posts,

WORDPRESS categories, products, comments, forms and other items. The general layout and look and feel of a site is determined by its theme and participants will learn how to find, choose and install a theme. Next attention is paid to the WordPress directory structure on the Web Server, the important configuration files there and how these files can be accessed using the File Transfer Protocol (ftp). A WordPress installation can be given additional functionality by the use of Plugins. Numerous plugins are available on the Internet, many are free, for others you will need to

buy a license. The finding and installation of plugins is discussed and attention is paid in particular to a Theme Plugin, an IP Blocker plugin, a Multi Language plugin and a plugin for Search Engine Optimalization (SEO).			
Module 1 : WordPress Intro	Module 2 : Using the Dashboard	Module 3 : Content types	
What is WordPress Content Management Systems Domain names	WordPress User Interface Using Dashboard Media and Content Administration	Pages and Tags Posts Versus Pages Organizing with Categories	

Content Management Systems Domain names Hosting Options Download WordPress Installing WordPress Directory Permissions Securing Passwords User Management	Using Dashboard Media and Content Administration WordPress Settings WordPress as Blog and Website Converting sites WordPress as CMS Other Uses	Posts Versus Pages Organizing with Categories Connecting Posts with Tags Custom Post Types Custom Taxonomies Managing Lists of Links WordPress Editors Image Editor Adding Video Media Adding Audio Media
Module 4 : WordPress themes	Module 5 : Configuration	Module 6 : Plugins
What are Themes? Structure of Themes Styling and CSS Installing Themes Configuring Themes Customizing Themes Theme Frameworks Parent-Child Themes Theme Best Practices	Server Directory Structure wp-config.php wp-login.php wp-admin wp-content wp-includes WP-DBManager Database Maintenance Common WordPress Problems Fixing Common Database Issues	Whats are Plugins? Finding Plugins Using Plugins Installing Plugins Upgrading Recommended Plugins Theme Plugin SEO Plugin Multi Language Plugin



## WordPress Web Development

DPRES

## PHP700: Course Joomla Web Development

Code: PHP700

#### Duration: 2 days

### Audience Joomla Web Development Course

This course targets persons that want to use Joomla for the design of a Web site or Web Application.

#### **Prerequisites Course Joomla Web Development**

No specific knowledge is required to participate in this course. General knowledge of Web applications and experience with computers is desired.

#### **Realization Training Joomla Web Development**

The theory is discussed on the basis of presentation slides. Demo's are used to clarify the treated concepts. In a number of subsequent exercises participants create a Joomla Website with often used functionality.

### **Course Joomla Web Development**

In the course Joomla Web Development, participants learn to use the Joomla Content Management System, CMS, to set up a complete web site. The course is based on the latest version of Joomla. Programming knowledge is no required to do this. It is discussed how Joomla components are put together and how they are linked. Attention is paid to components such as sections, articles, banners, ads, images and categories. Also addressed is how menus are linked in Joomla and how items and descriptions are added. Joomla can be extended with extensions that are available in





## **Joomla Web Development**





many places and that can extend the functionality. A number of these extensions will be demonstrated. Some are free for others you will need a licence. Further templates that give a certain structure to Joomla components are treated and application modules are discussed as well. Finally there is attention for all kind of issues related to the configuration of Joomla. After completing this course participants will be able to build a web site with Joomla all by themselves.

Module 1 : Joomla Intro	Module 2 : Content Types	Module 3 : Menu's
What is Joomla? How does Joomla work? Installing Joomla Frontend and Backend Home page Language setting Adding content Add an Article Add menu link	Sections and categories Creating sections Creating categories Assigning sections and categories Advertisements Making new pages Adding images Pagebreak Banners Contacts Enquête Search	Menu's Adding Menu-items Displaying sections Displaying categorylist Set details Add description Main Menu Change menu-items order Change level menu-item Change position submenu Removing menu-item More menu's External link Separation space
Module 4 : Joomla Templates	Module 5 : Modules	Module 6 : Configuration
What are Templates? Downloading Templates Installing Templates Changing Templates Using more template Change Logo Joomla template Mail Private message Group mail	What are modules? Adding a Module Adjusted HTML Login Syndicate Arbitrary image What are Parameters? Setting global parameters Setting individual parameters	General settings Tab Website Tab System Tab Server WYSIWYG-editor Frontend users Frontend registration Editing Account data Changing access level Backend users Access levels Media management

# PHP777: Course WordPress Theme Development

Code: PHP777

### Duration: 4 days

Module 1 : Intro WordPress Themes	Module 2 : Structure WordPress Themes	Module 3 : WordPress PHP Functions
What are Themes? Look and feel Theme alternatives Partial Adjustments CSS with Customizer Child themes	What are Templates? Template Files Template Hierarchy Basic Template Files Nested Templates Inner Templates	What are Template Tags? Anatomy of Template Tag Passing tag parameters Include tags Conditional tags Function Reference
Module 4 : WordPress Loops	Module 5 : Custom Post Types	Module 6 : Custom Fields

### Module 7 : Customizer

Sections and panels Settings and controls Several controls Input, Checkboxes, Color, Images etc. Data sanitization Refreshing the preview Retrieving customizer settings

## PHP800: Course Drupal Web Development

Code: PHP800

#### Duration: 2 days

### Audience Drupal Web Development Course

This course targets persons that want to use Drupal for the design of a Web site or Web Application.

#### **Prerequisites Course Drupal Web Development**

No specific knowledge is required to participate in this course. General knowledge of Web applications and experience with computers is desired.

#### **Realization Training Drupal Web Development**

The theory is discussed on the basis of presentation slides. Demo's are used to clarify the treated concepts. In a number of subsequent exercises participants create a Drupal Website with often used functionality.

### **Course Drupal Web Development**

In the course Drupal Web Development participants learn to use the Drupal Content Management System, CMS, to set up a complete web site. The course is based on Drupal version 7. To develop a Web Site in Drupal virtually no programming knowledge is required. Developers however can develop even more functional applications with the Drupal API. Next it is discussed how a typical site is built in Drupal and how the Drupal page model works. Participants learn the meaning and use of the various components in Drupal such as fields, views, modules, nodes,

blocks and pages. The course starts by installing Drupal and a first site is built and the associated modules are chosen. The course then continues with the layout with blocks and regions and also default blocks and custom blocks are discussed. The content types in Drupal including PAGES and ARTICLES on which fields of different types can be placed are discussed. Participants also learn what taxonomies are and how you can work with views, themes and input forms. Finally attention is paid to some advanced topics such as web services with XML-RPC. After completing this course participants can independently build a web site with Drupal.

Module 1 : Drupal Intro	Module 2 : Drupal Core	Module 3 : Layout and Files
What is Drupal? CMS Systems Drupal terminology Content Management Framework Web Application Framework Modules and Themes Nodes and Blocks Drupal Workflow Bootstrap Hooks and Callbacks Installing Drupal	Admin Interface Creating and Managing Content Site Building and Configuration User Management Out of the Box Modules Core Required Core Optional-enabled Core Optional-disabled User Contributed Modules Popular Modules Module selection and evaluation	Layouts in Drupal Blocks and Regions Default Blocks Custom Blocks Configuring Blocks Enabling Default Blocks Controlling the Front Page File System Download Methods File Module Image Module
Module 4 : Fields Module	Module 5 : Taxonomies	Module 6 : Drupal VIEWS
Custom Content Types The PAGE and the ARTICLE Input Filters Field Permissions Adding Fields to Content-Types Text Fields Numeric Fields Link and Image Fields Field Groups Node Reference Manage Display Settings Display Formats	What is taxonomy? Working with Taxonomy Vocabularies Required Vocabulary Controlled Vocabulary Single and Multiple Terms Adding Terms View Content by Term Storing Taxonomies Module-Based Vocabularies	Overview of VIEWS VIEW Types Default Views Overridden Views Normal Views Displays Basic Settings Display Types Basic Settings Fields vs Node Filters and Arguments
Module 7 : Themes	Module 8 : Forms	Module 9 : Advanced Topics
Theme System Architecture Theme Templates The .info file Theme Engine Hooks Creating a Theme Installing a Theme Theme Inheritance Modifying Base Themes Custom Stylesheets Overriding Theme Behavior	Forms with Webforms module The Form API Form Processing Validation Form Submission Redirection Creating Basic Forms Custom Module Basics Enabling the Custom Form Module Accessing the Custom Form Form API Properties	Rules with the Rules module Common Functions Relationships XML-RPC What is XML-RPC? XML-RPC Clients A Simple XML-RPC Server





## **Drupal Web Development**





## **PRG100: Course Programming Fundamentals**

Code: PRG100

#### Duration: 4 days

Price: € 1999

### **Audience Programming Fundamentals Course**

The course Programming Fundamentals is designed for individuals who want to learn the principles of programming and to apply their programming knowledge in any language on any platform.

#### **Prerequisites Course Programming Fundamentals**

To participate in this course no specific prior knowledge is required. General knowledge of and experience with computers is desirable.

#### **Realization Training Programming Fundamentals**

The theory is discussed on the basis of presentation slides. Demos are used to clarify the concepts. The C language is used as an example language. The theory is interspersed with exercises in which participants solve simple programming problems.

### **Certification Course Programming Fundamentals**

Participants receive an official certificate Programming Fundamentals after successful completion of the course.

### **Course Programming Fundamentals**



Memory Control Unit Arithmetic Logic Unit Accoundator

Input Output

## **Programming Fundamentals**





In the course Programming Fundamentals the basic principles of programming are discussed on the basis of a programming language. The language is not central to this course, but you need a language to program in. The method of structured programming is paramount. Simple problems are analyzed and converted into instructions in the programming language. The participants learn how to formulate a problem in Nassi Schneiderman diagrams and Data Flow diagrams. Attention is paid to writing programs in source code and translating this code with compiler and linker to executable binary code. The participants also learn about statements, operators, variables, constants, arrays and data types and control flow constructions such as branching with if, then, else, select, case and iterations with while, for, do, break and continue. Reuse of code with functions is discussed. Both calling functions from a library and writing functions yourself is subject matter of the course. The difference between call-by value and call-by reference is explained. Pointers which allow memory to be addressed and traversed are also discussed. The course ends with an introduction to object oriented programming. After completing the course, participants are able to write small programs that solve programming problems. They will not be aware of all the ins and outs of the language because that is not the purpose of this course.

Module 1 : Intro Programming	Module 2 : Structure Diagrams	Module 3 : Variables and Data Types
Programming Languages Language Syntax Levels of Programming Language Generations Unstructured Programming Procedural Programming Object Oriented Programming Compiled Language C Compiling and Linking Creating Executables Intermediate Language Java Compiler and Interpreter Compiling and Running Java Programs Script Language Python Running Python Scripts	Software Development Phases Structured Programming Pseudo Code Program Structure Diagrams PSD Instructions PSD Selection Diagrams Multiple Selection Diagrams Iteration Diagrams While and For Iteration Logical Operations in PSD's Input and Output I/O in PSD Average PSD File I/O in PSD Functions in PSD	Variables Data Types Assignment Instructions Variable Declaration Variable Initialization Java Data Types JavaScript Data Types JavaScript Data Types Identifiers Identifiers Examples Constants Strong Typing Weak Typing Dynamic Typing Comments
Module 4 : Control Flow	Module 5 : Operators	Module 6 : Arrays
Control Structures if Statement if else Statement if else Examples Multiple Selections Nested if Statements switch case Statement Iteration Statements for Loop while and dowhile Loop break and continue	What is an Operator? JavaScript Operators Arithmetic Operators Logical Operators Comparison Operators Assignment Operators String Operators Bitwise Operators Other Operators Operator Precedence Expressions	What are Arrays? Creating Arrays Initializing Arrays Accessing Arrays Array Indexes Array length Processing with for Processing with for each Multidimensional Arrays Associative Arrays JavaScript Associative Arrays PHP
Module 7 : Functions	Module 8 : Pointers	Module 9 : Classes and Objects
Library Functions User Defined Functions Calling Functions Advantages of Functions Function Prototype Function Definition Passing Parameters Local and Global Variables Return Statement Types of Calls Recursion	Pointers Variables and Addresses Pointer Declaration Initializing Pointers Pointers to Variables Pointer Dereferencing Pointer Assignment Call by Value and by Reference Pointers and Arrays Address Arithmetic Arrays in Function Calls	Class Definition Encapsulation Access Modifiers Constructors Creating Objects Fields and Methods Instance variables Class variables Using Objects Object References Object Destruction

## PRG200: Course C Programming

Code: PRG200

Duration: 5 days

### Audience Course C Programming

The course C Programming is designed for programmers who want to write programs in C or other people who want to understand C code.

### **Prerequisites Course C**

No prior programming knowledge is required to join this course. Knowledge of programming in another language is however beneficial to a quick understanding of the subject matter.

### **Realization Training C Programming**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Illustrative demo programs are used to explain the concepts further. The course material is in English.

### **Certification C Programming**

Participants receive an official certificate C Programming after successful completion of the course.

#### **Course C Programming**

In the course C Programming the basic principles of the C computer language are

covered. After an introduction about the background and characteristics of C and the function of the preprocessor, compiler and the linker, the program structure of C programs is discussed. Through a series of coordinated exercises, participants learn to program with the variables, data types, storage classes, operators and control flow constructs of the C language. Subsequently, the use of functions in C is addressed. It is explained how functions are declared and defined and how parameters are passed to functions. The difference between passing parameters by reference and by value is explained. Ample attention is also paid to the pointer concept, arithmetic with pointers, the equivalence between pointers and arrays and the use of function pointers. Next the focus is set on user defined composite data structures such as structures and unions. Finally, the C standard library is discussed, which includes the treatment of various functions for dealing with file IO, date and time, the manipulation of strings and the dynamic allocation of memory. Follow up courses for the course C Programming are Advanced C Programming and C++ Programming.

Module 1 : Intro C Language	Module 2 : Variables and Data Types	Module 3 : Control Flow
C Programming Language History of C C Characteristics C Programs Keywords Creating First C Program Compiling and Linking Preprocessor Header Files Creating Executables C Standard Library Basic I/O Comments C Compilers Resources	Variables Data Types in C Variable Names Data Type Sizes Signed and Unsigned Types Numeric Constants Character Constants String Constants Enumeration Constants Symbolic Constants Type Casting Arrays Multidimensional Arrays Character Arrays Derived Data Types	Control Flow Constructs Statements and Blocks If Statement If.else Statement Ambiguity of else else-if Construction switch Construction case Statements for Loop Nested for Loop while Loop do while Loop break And continue goto And Labels
Module 4 : Operators	Module 5 : Functions	Module 6 : Storage Classes
Arithmetic Operators In- and Decrement Operators Relational Operators Logical Operators Assignment Operators Bitwise Operators Shift Operators Bitwise Assignment Conditional Operators sizeof Operator Type Conversions	Library Functions User Defined Functions Calling Functions Function Prototype Function Definition Passing Parameters Call by Value Call by Value Call by Reference Local and Global Variables Return Statement Recursion	Storage Classes Automatic Variables External Variables Globals with Extern Static Variables Register Variables typedef Scope and initialization Initialization Array Initialization
Module 7 : Preprocessor Directives	Module 8 : Pointers	Module 9 : Structures and Unions
C Preprocessor #include Directive #define and #undef Directive Macro's with Arguments Macro Gotchas Conditional Inclusion #if #else #endif #elif Multiple Inclusion .h Files #ifdef and #ifndef Predefined Macros FILEandLINE	Variables and Addresses Pointer Declaration Initializing Pointers Pointers to Variables Pointer Dereferencing Pointer Assignment Pointers and Arrays Address Arithmetic null Pointer Pointers to Functions Character Pointers Command Line Arguments	Structure Definition Variables of Structure Type Accessing Structures Nested Structures Structure Initialization Pointers in and to Structures Structures and Functions Arrays of Structures Bitfield Structures typedef And Structures Unions Type Fields







dule 10 : Standard C Library
File I/O Functions Access Modes File Read and Write Function scanf Function Parameters Operation of scanf Utility Functions Flow Control Functions system Function ASCII to Binary Functions Environment Functions Memory Management Functions Time and Date Functions

## PRG202: Course Advanced C Programming

Code: PRG202

Duration: 3 days

### Audience Course Advanced C Programming

The course Advanced C Programming is intended for C developers who want to understand and use the more advanced features and techniques of the C language.

### Prerequisites Course Advanced C Programming

To join this course knowledge of the basic concepts of  $\underline{\textbf{programming in C}}$  and experience in C is required.

### **Realization Training Advanced C Programming**

The concepts are treated on the basis of presentation slides. Illustrative demo programs are used to clarify the treated concepts. There is ample opportunity to practice and theory and practice is interspersed.

### **Official Certificate Advanced C Programming**

Participants receive an official certificate Advanced C Programming after successful completion of the course.

### **Advanced C Programming Course**

In the course Advanced C Programming the more advanced aspects of C

programming like the use of modules and interfaces and object oriented programming in C using encapsulation are discussed. Potential dangers of certain constructs in C are highlighted such as C macros, evaluation order and the use of globals. Attention is paid to pointer techniques and the use of pointers to functions is discussed. The most common dynamic data structures such as arrays, linked lists, hash tables and trees and their implementation in C are addressed in detail. Also attention is paid to optimization techniques regarding speed and memory. More advanced functions of the C library such as setimp, longimp, signals, bsearch, and qsort etc. are also on the course program. Finally bit manipulation and advanced string handling and parsing are discussed.





## Advanced C Programming





Module 1 : C Pitfalls	Module 2 : Object Orientation in C	Module 3 : Pointers and Arrays
C Standards	Object Orientation	Pointers Revisited
Pitfalls C Language	Classes and Objects	Pointer Arithmetic
Write Clean Code	Example Class and Objects	Pointers and Arrays
Good Programming Style	Object Orientation in C	Function Pointers
Rules for Functions	Simple Objects in C	Optional Ampersand
Deep versus Flat Code	C versus C++	Variable Length Arguments
Evaluation Order	Class Constructor in C	Varargs Example
Avoid Macros	Member Functions in C	Array Initialization
Without const	Encapsulation in C	Array Traversal
Using const	Public Function Members in C	Arrays of Structs
Code Reuse	Inheritance	Arrays versus Pointers
Avoid Globals	Extending C++ Classes	Multidimensional Arrays
Pre and Post Conditions	Virtual Functions	Indices in 3d Array
Invariants	Virtual Function Table	Dynamic Arrays
Code Smells	Polymorphism	Ragged Arrays
Module 4 : Data Structures in C	Module 5 : Bit Manipulation	Module 6 : C Standard Library
Dynamic Data Structures	Bit Manipulation	What are Signals?
Singly Linked Lists	Bitwise Operators	ANSI C-Signal Types
Linked List Element Type	Bitwise AND	Handling Signals
Creating List Elements	Bitwise OR	SIGABRT, SEGV, SEGFPE
Doubly Linked Lists	Bitwise XOR	setimp and longimp
Stacks and Queues	Bitwise NOT	Coroutines
Hash Tables	Bitshift Operators	atexit Function
Load Factor	Bit Shifting	assert Function
Hash Functions	Bitwise Assignment Operators	perror Function
Trees	Bit Rotation	Raising Signals
Trees Traversal	BitFlag Functions	Alarm Signal

### Module 7 : String Handling

Looking for Characters Looking for Substrings Count Matching Characters Looking for Character Sets String Comparison String Tokenizing Convertion Strings to Numbers Handling Conversion Errors qsort and Bsearch Advanced String Handling

## PRG300: Course C++ Programming

### Code: PRG300

Duration: 5 days

TEMPI ATE

LIBRARY

### Audience C++ Programming Course

The course C++ Programming is intended for developers who want to learn programming in C++ and others who want to understand C++ code.

### Prerequisites Course C++ Programming

Knowledge of and experience with **C programming** is required to attend this course.

### **Realization Training C++ Programming**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Illustrative demos are used to clarify the discussed concepts. The course material is in English.

### **Certification C++ Programming**

Participants receive an official certificate C++ Programming after successful completion of the course.

#### **Course C++ Programming**

In the course C++ Programming participants learn to program in the C++ language. In the course the latest version of the C++ standard is used namely C++ 17. First the differences between C and C++ are discussed concerning variable declarations,

formatted output with the stream IO library, namespaces, function overloading and default function parameters. Subsequently the new C++ reference variables are discussed. Attention is paid to both Lvalue and Rvalue references. An important element of the course is the C++ class concept and C++ implementation of objectoriented principles such as abstraction and encapsulation. Attention is paid to dynamic memory allocation with new and delete and the role of assignment operators and copy and move constructors. Also special features of classes such as statics, friends and iterators are discussed. Next the object-oriented principles of inheritance and polymorphism are part of the subject matter. This includes the concepts of virtual functions, v-tables, dynamic binding and abstract classes. C++ has the option to give existing operators a different meaning and this phenomenon is discussed in the module operator overloading. Attention is paid to important features of the standard C++ library like the String class and the base concepts of C++ templates and the Standard Template Library (STL). Finally exception handling and how this is implemented in C++ is addressed. A follow up course for the course C++ Programming is <u>Advanced C++ Programming</u>.

Module 1 : Intro C++	Module 2 : Variables and Types	Module 3 : References
Intro C++	Standard Types	References
C++ TimeLine	Type Inference	Reference Initialization
Comments in C++	Auto Keyword	References and Pointers
Namespace std	Deduction with decltype	Rvalues and Rvalues in C
Output and Error Stream	Initialization	Rvalues and Rvalues in C++
Standard Input Stream	Null Pointer Constant	Reference to Constant
cin and Strings	Strongly Types Enums	Passing References
Formatted Output	Variable Scope	Comparison Parameter Passing
Variable Declaration	Namespaces	References as Return Values
Scope Resolution Operator	Using keyword and Directive	Returning Ivalue
Inline Functions	Block Usage	Returning Reference to Global
Default Function Arguments	User Defined Literals	Rvalue References
Overloading Functions	Storage Classes	Comparing Reference Types
Range based for loop	const Qualifier	Rvalue Reference Usage
Module 4 : Classes	Module 5 : Dynamic Memory Allocation	Module 6 : Inheritance
Classes and Objects	new and delete Operators	Inheritance
Classes in C++	Dynamic Arrays	Derived Classes in C++
Class Declaration	Classes with Pointer Data	Class Hierarchy
Class Sections	Assignment Operator	Redefining Member Functions
Constructor and Destructor	Self-Assignment Problem	Derived Class Constructors
Uniform Initialization	Chained Assignments	Base - Derived Class Conversion
Header and Sources Files	Assignment and Initialization	Pointer Conversions
Class Implementation	Copy Constructors	Virtual Functions
Advantages Access Functions	Passing Objects	Polymorphism
References to private Data	Returning Objects	Dynamic Binding
this Pointer	Passing References to Objects	Virtual Function Table
static Members	Move Constructor	Pure Virtual Functions
Constant Objects	Move Assignment Operator	Abstract Classes
Member Objects	Perfect Forwarding	Multiple Inheritance
Friends	Delegating Constructors	Virtual Derivation
Module 7 : Operator Overloading	Module 8 : Exception Handling	Module 9 : Templates
Operator Overloading	Exception Handling in C++	What are Templates?
Overloading for Numeric Types	Memory Exhaustion Handling	Template Functions
Complex Type Example	Throwing Exceptions	Template Specialization
Overloading Rules	try Block	Template Parameter List
Overloading Restrictions	catch Handlers	Class Templates
Not Overloadable Operators	Multiple catch Handlers	Template Parameter Scope
When not to Overload	Template Array Class	Template Function Statics
Numeric Class Overloading	Exceptions Array Class	Template Class Statics
Operators as Friend	catch Order	Inclusion Compilation Model
Unary Overloading Operator	throw List	Templates and Friends







STANDARD

TEMPLATE

LIBRARY
Module 10 : STL
Standard Template Library
STL Library Components
STL Containers
Vector Container
Deque Container
List Container
STL Iterators
STL Algorithms
STL Allocators

# PRG303: Advanced C++ Programmeren

Code: PRG303

# Duration: 3 days

Module 1 : Modern C++ Features	Module 2 : References and Move Semantics	Module 3 : Classes and Inheritance
C++ Timeline C++11 Features Type Inference Auto Keyword Deducion with decltype Deducing return Types Uniform Initialization Initializer Lists Range Based for Loop Null Pointer Constant constexpr Keyword Static Asserts Lambda Functions Lambda Functions Lambda Syntax Strongly Types Enums User Defined Literals Raw String Literals	Reference Initialization References and Pointers Rvalues and Rvalues in C++ Passing and Returning References Returning Ivalue Returning Reference to Global Rvalue References Comparing Reference Types Rvalue Reference Usage Assignment Operator Copy Constructor Passing and Returning Objects Passing References to Objects Move Constructor Move Semantics Move Assignment Operator Golden Rule of 5	default Keyword delete Keyword Delegating Constructors Inheritance Inheriting Members Calling Base Class Constructors Multiple Inheritance Virtual Derivation Polymorphism Virtual Functions Virtual Functions Virtual Function Table Abstract Classes Interfaces in C++ Destructors and Inheritance Virtual Destructors override Specifier final Specifier
Module 4 : Smart Pointers	Module 5 : Operator Overloading	Module 6 : Templates
unique_ptr Using unique_ptr Specialization for Arrays Replacement for std::auto_ptr std::make_unique shared_ptr Pointer Control Block shared_ptr Destruction Policy shared_ptr Interface Cyclic References std::enable_shared_from_this weak_ptr Using Smart Pointers	Operator Overloading Syntax Operator Overloading Overloading Numeric Types Overloading Overview Overloading Restrictions When not to Overload Operators as Class Members Operators as Friend Functions Overloading Stream Operators Overloading ostream Overloading istream Overloading Inary Operators Overloading Binary Operators	What are Templates? Template Functions Template Specialization Template Parameter List Inclusion Compilation Model Class Templates Template Member Functions Template Parameter Scope Templates and Statics Templates and Statics Recursive Analogy Alias Templates Perfect Forwarding
Module 7 : Exception Handling	Module 8 : Multiple Threads	Module 9 : Synchronization
Error Conditions and Exceptions Exception Handling Process Class Objects as Exceptions Parameter Catch Block Exception Hierarchy Catching in Hierarchy Golden Rule Retrowing Exceptions noexcept Specifier Preventing Resource Leaks RAII Idiom C++ Standard Exceptions User Defined Exceptions Exception Handling Costs	Multiple Threads Benefits and Drawbacks Thread Charactistics Thread Class Simple Threads Joining Threads Detaching Threads Thread ID Callables Passing Parameters Pass by Reference Pass by std::ref and std::move Member Function as Thread Thread Local Storage	Data Corruption and Synchronization Lock Guard Automatic Lock Management Mutex and RAII Recursive Locking Timed Locking Atomic Types Call Once Event Handling Condition Variables Wait and Notify Promises and Futures Asynchronous Tasks Working with async
Module 10 : C++14 and C++17 Features	Module 11 : Standard Template Library	
Init-statement for if Selection Initialization Structured Binding Types Structured Binding Declarations const if Expressions Guaranteed Copy Elision Inline Variables Nested Namespaces Fold Expressions Fold Expression Variations Optional Type Small String Allocations String View Generic lambdas Binary literals Aggregate initialization	STL Core Components Containers, Algorithms and Iterators Container classification Vectors, Lists and Dequeues Adapters Associative Containers Maps and Hash Maps Strings Bitsets STL Iterators Reverse iterators Iostream iterators Function objects STL Algorithms Predicates and Comparators STL Allocators	

# **PRG400: Course Python Programming**

Code: PRG400

Duration: 4 days

### **Audience Course Python Programming**

The Course Python Programming is intended for Developers and system administrators who want to learn how to program in Python and other persons who want to understand Python code.

#### **Prerequisites Python Programming Course**

Knowledge and experience with programming is not strictly required to participate in this course. Experience in programming is however is beneficial for a proper understanding.

### **Realization Training Python Programming**

The theory is treated on the basis of presentation slides. Illustrative demos are used to clarify the concepts further. The theory is interspersed with practical exercises. The course material is in English.

### **Certificate Python Programming**

Participants receive an official certificate Python Programming after successful completion of the course.

### **Course Python Programming**





# **Python Programming**





In the course Python Programming participants learn how to program in the object-oriented scripting language Python. Python is a language that is often used for installation scripts and for prototypes of large applications. After an introduction on the installation and the different ways to execute scripts, the basic concepts such as declarations, variables and control flow structures are discussed. Attention is also paid to collection structures, such as Lists, Tuples and Dictionaries. Next the focus is on the use of functions with the different methods of parameter passing, such as by value and by reference. The scope of variables and lambda functions are also discussed here. Subsequently attention is paid to the division of Python software into modules and the use of namespaces and packages. Comprehensions in Python library functions for accessing files is the subject matter and attention is paid to database access with the Python Database API. Also object-oriented programming with classes and objects is treated. In this respect concepts such as properties, constructors and encapsulation are highlighted. Finally if time allows, optional attention is paid to different libraries for Regular Expressions, unit testing and date and time. A follow up on this course is the course Advanced Python Programming.

Module 1 : Python Intro	Module 2 : Variables and Types	Module 3 : Data Structures
What is Python? Python Features Getting Started Setting up PATH Environment Variables Running Python Interactive Mode Script Mode Identifiers Reserved Words Lines and Indentation Multi Line Statements Quotes	Variables Data Types Python Numbers Numerical Types Number Type Conversions Conversion Functions Built-in Number Functions Python Strings String Operations String Formatting Triple Quotes Raw and Unicode Strings Built-in String Functions	Sequences and Lists Accessing and Updating Lists Multidimensional Lists List Operations List Functions and Methods Tuples Accessing Values in Tuples Tuple Functions Bytes and Byte Arrays Sets and Dictionaries Accessing Values in Dictionaries Properties of Dictionary Keys Dictionary Methods
Module 4 : Control Flow	Module 5 : Functions	Module 6 : Modules
Control Flow Constructs if Statement else Statement while Loop for Loop break Statement continue Statement Loop with else Combination pass Statement Python Operators Operator Precedence	Function Syntax Calling Functions Pass by Value Pass by Reference Overwriting References Function Arguments Keyword Arguments Default Arguments Variable Length Arguments Lambda Functions return Statement Scope of Variables	import Statement fromimport Statement Locating Modules Creating and Using Modules dir Function Python Packages Explict Import Modules Implicit Import Modules Namespaces and Scoping globals and locals Functions reload Function Test Harnass
Module 7 : Comprehensions	Module 8 : Exceptions	Module 9 : Python IO
Functional Programming Map and Filter Reduce and Lambda List Comprehensions Filtered List Comprehension Syntactic Sugar Dictionary Construction with Zip Dictionary Comprehension Dictionary from Keys Set Comprehension	Unexpected Errors Typed Exception Handling Exception Handling with Else except Clause Multiple Exceptions Standard Exceptions try-finally Clause Exception Arguments Raising Exceptions Example raising Exceptions User Defined Exceptions	Input and Output IO Module Opening Files File Open Modes Reading and Writing Binary Files Reading and Writing Text Files File Positions Renaming and Deleting Files Directory Methods Creating Directories
Module 10 : Database Access	Module 11 : Python Classes	Module 12 : Optional : Python Libraries

Python DB API Using with Inserting Data Prepared Statements Last inserted row id Retrieving Data Fetching Rows Parameterized Queries Transactions Object Orientation Creating Classes Class Members Creating and Using Objects Accessing Attributes Property Syntax Built-in Class Attributes Constructors and Destructors Encapsulation Regular Expressions match Function Matching versus Searching Search and Replace Unit Testing Unit Test Example Date and Time Handling Time Tuple Calendar Functions

# **PRG404: Course Advanced Python Programming**

Code: PRG404

Duration: 4 days

### Audience Course Advanced Python Programming

The course Advanced Python Programming is intended for Python developers who want to know more about the Python language and who wish to become proficient in advanced aspects of Python.

### Prerequisites Course Advanced Python Programming

To participate in this course knowledge of and experience with programming in Python is required.

### **Realization Training Advanced Python Programming**

The theory is discussed on the basis of presentation slides. Illustrative demos illustrate the concepts. The theory is interspersed with exercises. The course material is in English.

#### **Official Certificate Advanced Python Programming**

Participants receive an official certificate Advanced Python Programming after successful completion of the course.

#### **Course Advanced Python Programming**



# **Advanced Python Programming**





The course Advanced Python Programming discusses advanced aspects of the Python programming language that simplify and accelerate the development of Python software. In the first place a number of advanced aspects of classes are discussed such as multiple inheritance, polymorphism and operator overloading. Subsequently attention is paid to the use of modules and packages and participants learn how to create packages themselves, how to upload and install them in a virtual environment. Accessing XML and JSON data is also on the program and it is discussed how logging can be implemented in Python programs. Additionally iterators that deal with lazy evaluation are discussed as well as generators and coroutines with which concurrent programs can be made. Then decorators that are used to add functionality such as caching and proxying to existing functions and classes, are discussed. In the module patterns the Python implementation of different standard Design Patterns is explained. Next attention is paid to an advanced feature like meta programming. Also the creation of processes and threads is discussed as well as the synchronization between threads and optimizing the performance of Python code. Subsequently the new ayncio module is on the course program with which asynchronous IO with futures can be realized. Interprocess communication via sockets and pipes is also treated. And finally unit and mock testing is discussed in the context of test automation.

Module 1 : Advanced Classes	Module 2 : Modules and Packages	Module 3 : XML en JSON Access
Classes Recapitulation Data Hiding Property Syntax Inheritance super Keyword Multiple Inheritance Constructor Chaining Checking Relationships issubclass and isinstance Overriding Methods str andrepr Class Methods Operator Overloading Polymorphism	import Statement from import Statement Locating Modules Packages in Python Explicit and Implicit Import Namespaces and Scoping Test Harnas Virtual Environments and Activation Distribution of Packages Installing packages pip install Using Python Package index PyPI commands Uploading Package with Setup	XML Parsing Pull versus Push Parsing Python XML Libraries DOM and SAX DOM Navigation and Manipulation XPath Minidom ElementTree Reading and Writing XML Searching and Validating XML XML Manipulation JSON library Dictionary to JSON conversion Loading and Dumping JSON
Module 4 : Logging	Module 5 : Generators	Module 6 : Decorators
logging Module When Use Logging Log Levels Logging Configuration Log in Multiple Modules	Iteration Iterables Iteration Protocols Supporting Iteration	Functions as Objects Passing and Returning Functions What is a Decorator? Decorator Syntax
Formatting Logging Logging Components Logger per Module Handlers and Filters Logging Flow Formatting Logger Adapter	Generators Generator Functions Conveniant Iterator Generator Expression Expression Syntax Building Blocks Chaining generators Coroutines	Types of Decorators Passing Arguments Multiple Decorators Class Decorators Singleton Class Why Decorators Need for AOP Crosscutting Security Concern

What are Patterns?	Classes as Objects	Thread Characteristics
Singleton Pattern	Metaclasses	Threads in Python
Adapter Pattern	Object from Metaclass	Current Thread
Chain of Responsibility Pattern	Class of Class	Daemon Threads
Observer Patters	Descriptor Protocol	Joining Threads
Patterns or Principles	Lookup	Derived Thread Class
Everything is Object	Property	Signaling Threads
EAFP	Functions and Methods	Lock Object
Duck Typing	Classes and Types	Locks as Context Managers
Monkey Patching	Object Creation	Condition Synchronization
Dependency Injection	Metaclass	Barriers
None	Singleton As Type	Semaphores
Context Managers	Object Construction	Thread Local Data
Module 10 : Async IO	Module 11 : Networking	Module 12 : Unit Testing
Concurrent Execution	Network Layering	What is Unit Testing?
Multiprocessing	TCP/IP Layering	Automated Testing
Subprocess	UDP versus TCP	Test Driven Development
Scheduler	TCPv4 versus TCPv6 sockets	Traditional versus TDD
Queue	Connectionless Services	Unit Testing in Python
AsynIO	Connection Oriented Services	Python Test Frameworks
Task	Socket Utility Functions	Test Cases
Future	Asynchronous Servers	Assertions
Concurrent Futures	Using Pipes	Fixture
Eventloop	Anonymous and Named Pipes	Test Suite

# PRG440: Cursus C++ 11 Overview

Code: PRG440

### Duration: 1 day

Module 1 : C++11 Intro	Module 2 : Useability Enhancements	Module 3 : Functionality Improvements
C++ Timeline C++ 11 Features Type Inference Auto Keyword Deduction with decltype Deducing return Types Initialization Uniform Initialization Initializer Lists Initialization Priorities Range Based for Loop Null Pointer Constant Standard Types Strongly Types Enums User Defined Literals Raw String Literals C++11 Compiler Support C++14 Features	RValue References LValues and RValues in C LValues and RValues in C++ RValue Reference Syntax Comparing Reference Types RValue Reference Usage std::move Move Constructor Move Semantics Move versus Copy Constructor Move Assignment Operator Perfect Forwarding Delegating Constructors	Lambda Functions Lambda Syntax Lambda Usage constexpr Keyword override Specifier final Specifier Static Asserts Variadic Templates Recursive Analogy Alias Templates Smart Pointers auto_ptr auto_ptr Ownership shared _ptr Destruction Policy shared_ptr Interface Cyclic References unique_ptr weak_ptr

# Module 4 : Standard Library Changes

New Thread Library Launching Threads ThreadID's Detaching Threads Callables as Thread Function Data Corruption Synchronization Mechanisms Mutex Deadlock Automatic Lock Management Threading API Promises and Futures Regular Expressions Random Numbers Time Utilities New Containers

# PRG450: Course Django Web Development

Code: PRG450

#### Duration: 3 days

# Audience Django Web Development Course

The Django Web Development course is designed for developers who want to use the Framework for creating Web Applications in Python.

#### Prerequisites Course Django Web Development

To participate in this course knowledge of Web Applications and knowledge of and experience with **programming in Python** is required.

#### **Realization Training Django Web Development**

The theory is treated on the basis of presentation slides. Demos are used to clarify the concepts further. During the course there is ample opportunity to practice. The course material is in English. The course times are from 9.30 up and to 16.30.

### **Certification Course Django Web Development**

Participants receive an official certificate Django Web Development after successful completion of the course.

#### **Course Django Web Development**

Customizing

Resources

In the course Django Web Development participants learn how to use this Web Application Framework for developing Python Web Applications. The course starts

with an overview of the architecture of the Framework, how it can be installed and how the Admin application can be used. Next the MVT pattern is treated and how Models are mapped to the database. Also the various Field types and the data access API are discussed. Next attention is paid to views and URLs and it is shown how regular expressions are used in the mapping to views. Also the use of templates in the construction of views is discussed. Then the focus of the course turns to the creation of Forms used to collect input from the user. Attention is also paid to the validation of Forms and the use of the so-called Model Forms derived from Model classes. Object Relational Mapping and performance optimization when accessing the database is discussed as well. Finally some more advanced topics are on the program such as authentication in Django, working with the REST Services and Unit Testing. By the end of the course the participants will have built a complete Application that includes a REST interface.

Django

Module 1 : Django Intro	Module 2 : Django Models	Module 3 : Views and URL's
What is Django? Django History Framework Features Python CGI Script MVC Design Pattern Creating Projects Project Settings Project URL's Running Project Testing Project Admin Application Setup Databases Activate Admin Site	Mapping Models Create Application Create Models Migrations SQL for Models Three Step Migration Practice Data Access API String Representation Fields Creating and Accessing Objects Enabling Admin Interface Customize Admin Form Adding Related Objects	View Basics URL Mapping Django MVT Pattern URL Arguments URL Utility Functions Non-Named Group Matching Named Groups View Functions Mapping to Views HTTP Request Object HTTP Response Object redirect Shortcut get_object_or_404
Module 4 : Django Templates	Module 5 : Django Forms	Module 6 : Django Security
The Template System Template Variables render_to_repsonse Shortcut render Shortcut Context Variable Lookup List in Template Template Tags If and For Tag Filters Template Inheritance Child Templates Autoescape Loading Templates Class Based Views Specialized Views	Form Objects Using Form in View Processing Form Data Display Form using Template Display using Paragraphs Display using Table Validating Forms Customize Form Template Rendering Error Messages Looping over Form Fields Rendering Forms Core Fields Argument ModelForms Model and Form Customize Model Forms	Web Security Essentials Bad Practices Web Security Security Features Cross Site Scripting XSS Protection Cross Site Request Forgery CSRF Protection Clickjacking Protection SQL Injection Protection Cookies and Files Email Header Injection Django Authentication Authenticating Users Permissions and Authorization Authentication in Web Requests
Module 7 : Django REST	Module 8 : Advanced Topics	
What is REST? REST Services REST Examples Resource URI's REST challenges Django Solutions Includes Installing DRF Core Components Django Counterparts Building our Demo API	Sessions Session Support Messages Framework Using Messages Sending Email Emitters Testing Unit Testing ORM advanced Aggregation and annotation Reporting/Data analysis application	

Database performance profiling

Signals to denormalise



# Django Web Development

URL Mapping

Filesystem

Database

# PRG460: Course Rust Programming

Code: PRG460

Duration: 3 days

### Audience Course Rust Programming

The course Rust Programming is intended for developers who want to learn how to program in Rust and others who want to understand Rust code.

### **Prerequisites Course Rust Programming**

Experience programming in a modern programming language is desirable and beneficial to a good understanding.

#### **Realization Training Rust Programming**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

# **Certification Rust Programming**

The participants receive an official certificate Rust Programming after successful completion of the course.

#### **Course Rust Programming**

In the course Rust Programming participants learn to develop software with the latest version of the innovative programming language Rust. Rust is a new, practical system

programming language that produces lightning fast code. Rust is community driven. With Rust you prevent almost all crashes and data races. Rust builds on a rich history of programming languages. It is low-level language with multiple paradigms, both imperative and functional and Rust focuses on safe, high-performance, concurrent applications. Rust began to gain momentum in the industry before the official 1.0 version in May 2015, because there is a clear need for a new low-level system language. This course deals with what makes Rust so unique and applies this to practical problems of system programming. Topics that will be discussed are: traits, generics, memory safety, move semantics, borrowing and lifetimes. And also the rich macro-system of Rust, closures and concurrency are discussed.

Module 1 : Rust Intro	Module 2 : Variables and Data Types	Module 3 : Expressions and Flow Control
What is Rust?	Primitives	Expressions
Rust Background	Tuples and Arrays	Flow Control
Rust Momentum	Slices	if else
Rust Usage	Custom Types	loop
Comparisons to C	Enums	Nesting and labels
Rust Applications	Constants	while
Hello Rust	Variable Bindings	for and range
Comments	Scope	match
Formatted Printing	Shadowing	Guards
Debug and Display	Casting	Binding
Literals	Inference	if let
Operators	Alias	while let
Module 4 : Functions	Module 5 : Modules	Module 6 : Generics
Methods	Visibility	Functions
Closures	Struct Visibility	Implementations
Capturing	use Declaration	Parametrization over Types
As Input Parameters	Using super	Traits
Input Functions	Using self	Bounds
Type Anonymity	File Hierarchy	Multiple Bounds
As Output Parameters	Crates	Where Clauses
Examples from std	Attributes	Associated Items
Iterator::any	Extern crate	Associated Types
Iterator::find	Dead Code	Phantom Type Parameters
Higher order Functions	Custom	Unit Clarification
Module 7 : Scoping and Lifetimes	Module 8 : Traits	Module 9 : Standard Library
RAII	Zero cost Abstraction	Box, stack, heap
Ownership and Moves	Traits are interfaces	Data Structures
Functions and Methods	Derive	Vectors
Mutability	Operator Overloading	Strings
Borrowing and Freezing	Drop	Hashmap
Aliasing	Iterators	Threads Channels
ref Pattern	Clone	Path
Lifetimes	Designators	File I/O
Explicit Annotation	Overload and Repeat	Pipes
Bounds and Coercion	Unsafe Operations	Wait
Static	Static dispatch	Arguments
Elison	Dynamic dispatch	Meta



# **Rust Programming**

59.30

Multithreaded programming

# PRG500: Course Modern C++ Programming

Code: PRG500

#### Duration: 4 days

### Audience Modern C++ Programming Course

The course Modern C++ Programming is intended for developers who are familiar with <u>C++</u> but who wish to delve into the newest and most advanced techniques and features of the language.

#### Prerequisites Course Modern C++ Programming

To join this course knowledge of the basic concepts of <u>C++</u> and programming experience in C++ is required.

#### **Realization Training Modern C++ Programming**

The concepts are treated on the basis of presentations. Illustrative demo's are used to clarify the concepts. Theory and practice are interspersed and there is ample opportunity to practice.

#### **Certification Modern C++**

Participants receive an official certificate Modern C++ Programming after successful completion of the course.

#### Course Modern C++ Programming

In the course Modern C++ the new and advanced aspects of the C++ language based

on the standards C++11, C++14 and C++17 are extensively discussed. The course starts with an overview of the features that have been introduced in C++11 such as type inference, initializer lists, range based for loop, lambda functions and strongly typed enums. Next right references are discussed and the performance gains that can be booked by using move constructors in addition to copy constructors. The new possibilities with respect to inheritance with the keywords default, delete, override and final are also on the course program. In passing by the implementation of virtual functions and the need for virtual destructors are discussed. Attention is paid to smart pointers like unique pointers, shared pointers and weak pointers. Operator overloading and templates are also on the program and variadic templates and perfect forwarding are treated. The modern C ++ Resource Acquisition is Initialization or RAII idiom is explained while discussing exception handling. Attention is also paid to threads as well as the synchronization between threads which are now part of the C++ standard. This also includes asynchronous calls with promises and futures. Next specific C++11 and C++17 features are discussed such as optional types, structured binding declarations and constructions from the world of functional programming such as fold expressions. The course concludes with an overview of the advanced features of the Standard Template Library <u>STL</u>.

Module 1 : Modern C++ Features	Module 2 : References and Move Semantics	Module 3 : Classes and Inheritance
C++ Timeline C++11 Features Type Inference Auto Keyword Deduction with decltype Deducing return Types Uniform Initialization Initializer Lists Range Based for Loop Null Pointer Constant constexpr Keyword Static Asserts Lambda Functions Lambda Syntax Strongly Types Enums User Defined Literals Raw String Literals	Reference Initialization References and Pointers Rvalues and Rvalues in C++ Passing and Returning References Returning Ivalue Returning Reference to Global Rvalue References Comparing Reference Types Rvalue Reference Usage Assignment Operator Copy Constructor Passing and Returning Objects Passing References to Objects Move Constructor Move Semantics Move Assignment Operator Golden Rule of 5	default Keyword delete Keyword Delegating Constructors Inheritance Inheriting Members Calling Base Class Constructors Multiple Inheritance Virtual Derivation Polymorphism Virtual Functions Virtual Functions Virtual Functions Virtual Function Table Abstract Classes Interfaces in C++ Destructors and Inheritance Virtual Destructors override Specifier final Specifier
Module 4 : Smart Pointers	Module 5 : Operator Overloading	Module 6 : Templates
unique_ptr Using unique_ptr Specialization for Arrays Replacement for std::auto_ptr std::make_unique shared_ptr Pointer Control Block shared_ptr Destruction Policy shared_ptr Interface Cyclic References std::enable_shared_from_this weak_ptr Using Smart Pointers	Operator Overloading Syntax Operator Overloading Overloading Numeric Types Overloading Overview Overloading Restrictions When not to Overload Operators as Class Members Operators as Friend Functions Overloading Stream Operators Overloading ostream Overloading istream Overloading Unary Operators Overloading Binary Operators	What are Templates? Template Functions Template Specialization Template Parameter List Inclusion Compilation Model Class Templates Template Member Functions Template Parameter Scope Templates and Statics Templates and Statics Templates and Friends Recursive Analogy Alias Templates Perfect Forwarding
Module 7 : Exception Handling	Module 8 : Multiple Threads	Module 9 : Synchronization



C++98

contain algorith Strings C++11

C++14

C++17

ture

Error Conditions and Exceptions Exception Handling Process Class Objects as Exceptions Parameter Catch Block Exception Hierarchy Catching in Hierarchy Golden Rule Retrowing Exceptions noexcept Specifier Preventing Resource Leaks RAII Idiom C++ Standard Exceptions User Defined Exceptions Exception Handling Costs	Multiple Threads Benefits and Drawbacks Thread Charactistics Thread Class Simple Threads Joining Threads Detaching Threads Thread ID Callables Passing Parameters Pass by Reference Pass by std::ref and std::move Member Function as Thread Thread Local Storage	Data Corruption and Synchronization Lock Guard Automatic Lock Management Mutex and RAII Recursive Locking Timed Locking Atomic Types Call Once Event Handling Condition Variables Wait and Notify Promises and Futures Asynchronous Tasks Working with async
Module 10 : C++14 and C++17 Features	Module 11 : Standard Template Library	
Init-statement for if Selection Initialization Structured Binding Types Structured Binding Declarations const if Expressions Guaranteed Copy Elision Inline Variables Nested Namespaces Fold Expressions Fold Expression Variations Optional Type Small String Allocations String View Generic lambdas Binary literals Aggregate initialization	STL Core Components Containers, Algorithms and Iterators Container classification Vectors, Lists and Dequeues Adapters Associative Containers Maps and Hash Maps Strings Bitsets STL Iterators Reverse iterators Iostream iterators Iostream iterators Function objects STL Algorithms Predicates and Comparators STL Allocators	

# PRG500: Course Ruby Programming

Code: PRG500

#### Duration: 3 days

Price: € 1499

# Audience Ruby Programming Course

The course Ruby Programming is intended for experienced developers who want to learn the Ruby scripting language.

#### Prerequisites C, C++, C#, Visual Basic, Java or Perl

Knowledge and experience with programming in another programming language, such as C, C++, C#, Visual Basic, Java or Perl is required.

#### **Realization Training Ruby Programming**

The theory is discussed on the basis of presentation slides and is interspersed with practical exercises. Illustrative demos provide further clarification of the concepts. The course material is in English.

# **Certification Ruby Programming**

Participants receive an official certificate Ruby Programming after successful completion of the course.

#### **Course Ruby Programming**

In the course Ruby Programming participants learn to program in Ruby using the syntax and language constructs of that language. After an introduction about the

background of Ruby, the installation and the way in which code can run, the variables, data types and control flow are discussed. Attention is also paid to methods and parameter passing in Ruby and to object oriented aspects such as classes, objects, inheritance and polymorphism. The handling of errors through exception handling is discussed as well. The course continues with a discussion of partitioning code into modules, modules as namespaces and mixins as the use of modules from the standard library. Also closures such as blocks, lambdas and Procs are discussed and Ruby meta programming is treated with introspection, Open Classes and dynamic method invocation. Finally writing RubyGems is on the course schedule and an overview of the capabilities of the Standard Library is given.

Module 1 : Ruby Intro	Module 2 : Variables and Data Types	Module 3 : Control Flow
What is Ruby? Ruby Timeline Object Orientation Installation Interactive Ruby Ruby Execution Loading Ruby Code Naming Conventions Executing External Programs Ruby Blocks Resources	Numbers and Big Numbers Strings and String Literals String Interpolation Arrays Hash Range Struct Types of Variables Naming Conventions Constants Pre-defined Variables	Statements Assignment operators Conditionals Multiple Selection while and until Loop for Loop each Iteration Aritmetic operators Comparison operators Ruby truth Logical Operators
Module 4 : Methods and Parameters	Module 5 : Classes and Objects	Module 6 : Exception Handling
Method Definitions Invoking Methods Methods and Parenthesis Return values Default value argument Variable Argument List Array Argument Hash Argument Methods with Code Block Method with Bang Aliasing Methods	Classes and Objects in Ruby Object Initialization Attribute Accessors Current Object Class Variables and Methods Method Visibility Singleton Methods Inheritance Overriding Method Lookup Duck Typing	Error Handling Exception Handling Raising Exceptions Handling Exceptions Exception Class Hierarchy Typed Exception Handling Ensure Block Retry Command Throw and Catch Raising Exceptions User Defined Exceptions
Module 7 : Modules	Module 8 : Closures	Module 9 : Meta Programming
Modules Module Importing Files without Namespacing Modules for Namespacing Namespaces Mixins Mixin Example Include versus Extend Mixins and Inheritance Chain Modules versus Classes Comparable Module Enumerable Module	Benefits of Closures Lambdas and Procs Lambdas with Parameters Procs versus Lambdas Proc as Argument and Return Proc Objects as Closure What are Blocks? Representing Blocks Calling Blocks with Yield Passing Arguments Ampersand Operator From Proc to Block	What is Introspection? Introspection Code Classes are Open Class Definition are Executable Receivers Classes are Objects Dynamic Method Invocation Method Missing Invoking method_missing define_method Evaluating Code Dynamic Typing
Module 10 : Ruby Gems	Module 11 : Ruby Standard Library	





# Ruby Programming





What are Gems? Creating Gems gemspec file Installing and Using the Gem Publish the Gem Grabbing the Gem Rake Writing Tests Documenting Code Using Rdoc Using Bundler

Standard Library Overview Files Accessing Files File Open Modes Reading and Writing Directories Date and Time XML Access DOM Parsing SAX Parsing MultiThreading

# PRG505: Course Ruby on Rails Programming

Code: PRG505

#### Duration: 3 days

**Price:** € 1499

### Audience Ruby on Rails Programming Course

This course is intended for developers who want to work with the new state of art framework Ruby on Rails or Rails for short.

#### Prerequisites Course Ruby on Rails Programming

Some basic programming experience and knowledge is required. Knowledge of web standards such as HTTP and HTML is an advantage.

#### **Realization Training Ruby on Rails Programming**

The theory is discussed on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

# **Certification Ruby on Rails Programming**

Participants receive an official certificate Ruby on Rails Programming after successful completion of the course.

#### **Course Ruby on Rails Programming**

In this Rails training the participants are trained to become Rails developers. Rails is a framework for the pragmatic development of a full-blown web application. The

framework is based on the Model Driven Design paradigm. First the most important characteristics of the Ruby programming language are discussed. This knowledge is needed to be able to develop without too many problems with Rails. After this introduction, attention is entirely focused on the Rails framework. The whole Rails stack (model, view and controller) is treated in full detail. Then all the important issues for the successful build a Rails application are on the agenda such as Active Record, Action View and Controller.

Module 1 : Rails Intro	Module 2 : Rails and REST	Module 3 : Active Record
Introducing Ruby Installing Ruby on Rails Rails Environments and Configuration Rails Routing	REST (Representational State Transfer), Resources and Rails Controllers	Active Record ActiveRecord Migrations Active Record Associations Validations Advanced Active Record
Module 4 : Rails Actions	Module 5 : Security	Module 6 : Performance
Action View Helpers (Advanced) Rails and AJAX	Session Management Authentications and UserControl XML and Active Resource Action Mailer	Caching and Performance (Advanced) Logging and Troubleshooting RSpec and Rails API Plugins Batch and Background Processing





# **Ruby on Rails**





# **PRG550: Course Functional Programming**

Code: PRG550

Duration: 1 day

### Audience Functional Programming Course

The course Functional Programming is designed for individuals who want to learn the principles of functional programming and to apply this knowledge in languages with functional aspects.

#### **Prerequisites Course Functional Programming**

To participate in this course experience with software development and basic programming skills are required.

#### **Realization Training Functional Programming**

The theory is discussed on the basis of presentations. Demos are used to clarify the concepts. Examples in different languages are used to illustrate the concepts. The theory is interspersed with exercises.

#### **Certification Course Functional Programming**

Participants receive an official certificate Functional Programming after successful completion of the course.

#### **Course Functional Programming**

In the course Functional Programming the basic principles of functional programming

are discussed and illustrated with examples in different languages. In the functional programming paradigm, a program is built up of functions that perform operations on the arguments. Programming languages that are based entirely on this paradigm are Erlang, F<sup>#</sup>, <u>Haskell</u>, Lisp and ML. Modern languages such as Java, Python, <u>Scala</u> and JavaScript are equipped with functional features but are not entirely based on them. In purely functional programming languages functions cannot cause side effects. Side effects influence more than the result of the function like changing a global variable. Variables that serve as accumulator, counter or control variable do not belong in a functional language. The course discusses the backgrounds of functional programming in lambda calculus. Attention is also paid to characteristic functional concepts such as immutable data, clojures and map and reduce algorithms. The various types of functions such as first class functions, higher order functions and lambda functions are discussed. Finally concepts such as Generators and Iterators, Fold Expressions, Memoization and Monads are on the course program.

Module 1 : Intro Functional Programming	Module 2 : Functions and Style	Module 3 : Map and Reduce
What is Functional Programming? Paradigm Origins Lambda Calculus Absence of side effects Immutable Data and Keeping State Maintenance Benefits Functional Languages Erlang an Clojure Haskell and JavaScript Support in Python and Java	Functional Style Imperative Programming Declarative Programming First Class Functions Functions as Parameters Functions as Return Values Pure Functions Anonymous Functions Higher Order Functions Recursion	Replace Iteration Map Parameters Function and Collection Anonymous Functions Lambda's Reduce to Single Value Reduce Parameters Benefits Map and Reduce Filter and Find Function All and Any Function
Module 4 : Pipelines	Module 5 : Other Functional Concepts	
Combining Functions Parallelism Composition Efficiency Issues Referential Transparency Tail Call Optimization Currying Lazy Evaluation Determinism	Generators and Iterators Fold Expressions Memoization Monads Strict Evaluation Type Systems Non-strict Evaluation Evil of Shared State Thread Safeness	





# Functional Programming



# **PRG600: Course Perl Programming**

Code: PRG600

Duration: 3 days

# Audience Perl Programming Course

The course Perl Programming is intended for System administrators, Web masters and developers who want to learn to program in Perl or who want to understand Perl code.

### **Prerequisites Course Perl Programming**

Knowledge and experience with programming is not strictly required to participate in this course but is beneficial to a proper understanding.

#### **Realization Training Perl Programming**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Illustrative demo programs further clarify the concepts covered. The course contents covers the topics of the CIW Perl exam (1D0-437). The course material is in English.

### **Certificate Perl Programming**

Participants receive an official Perl Programming certificate after successful completion of the course.

### **Course Perl Programming**





# **Perl Programming**





In the course Perl Programming the principles of the Perl programming language are discussed. After an introduction on the characteristics of Perl, Perl modules and the typical usages of Perl, it is shown how a typical Perl script is executed. Subsequently, attention is paid to the syntax and use of scalar types, variables, operators and control structures. The use of Perl data structures like arrays, lists and hashes is also discussed. Part of the course is also the interaction between scripts and input and output devices such as keyboard and console and also attention is paid to the dealing with files and directories. In this respect the diamond and chomp operator are discussed. Using subroutines in Perl is part of the subject matter as well. On the basis of a number of structured exercises, participants learn how regular expressions in Perl can be used to validate data and to search for specific string patterns. Finally, an introduction to object oriented programming in Perl is given and it is discussed how additional Perl modules can be installed and used. Several Perl modules can be used as examples such as modules for Graphical User Interfaces or database access. Interspersed in other subjects the debugging of Perl scripts with the Perl debugger is also treated.

Module 1 : Perl Introduction	Module 2 : Scalar Types and Variables	Module 3 : Control Flow
What is Perl? Perl Characteristics Hello World in Perl Typical Perl Script strict Pragma Perl Modules Installation Perl IDE's Interactive Perl Resources for Perl Usages of Perl Perl Script reading Input Perl CGI Script	Scalar Variables Numbers and Numeric Operators Strings and String Literals String Number Conversions Scalar Variable Interpolation Reading from STDIN Comparison Operators if and while control structures Boolean Values chomp Operator undef Value and defined Function Special Scalar Variables References and dereferencing	Conditional Statements if elsif and else unless and unless else Logical AND and OR Conditional Expression Operator given when Loop Statements while and dowhile until and dountil for and foreach Jump Statements next and last redo and goto
Module 4 : Lists and Arrays	Module 5 : Subroutines	Module 6 : Input and Output
What are Arrays and Lists? Accessing Array Elements Array Operations Special Array Indices qw Shortcut List Assignment pop, push, shift and unshift sort and reverse Interpolating Arrays into Strings foreach Control Structure Default Variable \$_ Scalar and List Context Multidimensional Arrays	Subroutines Defining a Subroutine Invoking a Subroutine Return Values Arguments Private Variables my Variables Variable-Length Parameter Lists return Operator Non Scalar Return Values State Variables More on Parameter Passing Another Subroutine Example	Reading from STDIN Input from Diamond Operator Invocation Arguments Formatted Output Arrays and printf File Handles Reading from a File Changing Default File Handle Reading Whole File File Tests Operators Manipulate Files/Directories Listing Directories Executing External Programs
Module 7 : Hashes	Module 8 : Regular Expressions	Module 9 : Perl Modules
What is a Hash? Benefits of Hashes Hash Element Access Hash as a Whole Hash Assignment More Hash Syntax Hash Element Interpolation Hash Functions More Hash Functions The %ENV Hash Counting with Hash Merging Hashes	Simple Patterns Meta Characters Quantifiers and Character Classes Regular Expression Delimiters Subgrouping and Backreferences Regular Expression Modifiers split and join Named Captures Named Backreferences Automatic Match Variables Substitutions with s/// Binding Operator	What are Perl Modules? Packages my versus our Variables Module Naming Module Namespace Creating and Using Modules Modules in Subdirectories Accessing Module Variables Exporting from Modules Importing Modules Recommended CPAN Modules

# Module 10 : Object Orientation

Object Oriented Programming Object Oriented Programming in Perl Classes and Objects Example Class and Objects Class Constructor Properties or Fields Methods and Accessors Using Objects Inheritance Inheritance with @ISA Array Overridden Methods

# PRG606: Course Advanced Perl Programming

Code: PRG606

Duration: 3 days

### Audience Advanced Perl Programming Course

This course is intended for experienced Perl programmers who want to learn advanced features of the Perl language.

#### Prerequisites Course Advanced Perl Programming

Knowledge of and experience with programming in the Perl programming language is required to participate in this course.

#### **Realization Training Advanced Perl Programming**

The subjects are discussed on the basis of presentation slides. Demos help to clarify the concepts further. The theory is interspersed with exercises. The course material is in English.

# **Course Advanced Perl Programming**

This Advanced Perl Programming course is designed for experienced Perl programmers who want to explore the many advanced topics available. The Perl language has evolved into a very powerful multi purpose language. Modules written for Perl allow programmers to quickly create programs that incorporate complex data types, object-oriented technologies, graphical user interfaces and more. The subjects on the schedule of this course include the usage of Perl packages, the creation of Perl modules, using CPAN and accessing databases with Perl.





# Advanced Perl Programming



be \$mine for Geven

Module 1 : Perl Refresher	Module 2 : Debugging Perl	Module 3 : List Manipulation
Miscellaneous functions Diverse operators grep(), map(), qw// Reference data type Complex data types Advanced regular expressions Modules	Debugging Perl Warnings Diagnostic Messages Carping, Confessing, and Croaking Strict Checks Compiler Pragmas Debugging Flags Your Perl Configuration The Devel::Peek Module The Data::Dumper Module	Expert List Manipulation in Perl The grep Operator Lists, Arrays, and List Operators Context Context and Subroutines Initializing Arrays and Hashes Reference Syntax Auto-vivification Defined Values Other List Operators Usage of map, grep, and foreach
Module 4 : Blocks and Code References	Module 5 : Perl Packages	Module 6 : Perl Object Orientation
Blocks Subroutines Subroutine Prototypes Code Refs and Anonymous Subroutines Typeglobbing for the Non-Squeamish Local (Dynamic) Variables Lexical Variables Persistent Private Subroutine Variables Closures The eval Operator The Block Form of eval Block Form of eval for Exception Handling	Perl Packages Review of Packages BEGIN and END Blocks Symbol Tables Package Variables Calling Package Subroutines Importing Package Symbols Exporting Package Symbols Using the Exporter Package The use Function AUTOLOAD and @ISA AutoLoader and SelfLoader	Objects and Classes in Perl Object-Oriented Stuff Making Perl Object-Oriented References The bless Function So, What's a Blessed Thing Good For? Calling Class and Object Methods Object Methods Writing Classes Constructors Inheritance What Perl Doesn't Do
Module 7 : Tied Variables	Module 8 : Perl Modules	Module 9 : Perl Data Access
Tied Variables in Perl Why Use tie? Tying a Scalar Inside Tied Variables untie Another Tied Scalar Example Tying an Array A Tied Array Example Tying Hashes Tie::Hash and Tie::Array Tying Filehandles What Are DBM, NDBM, GDBM, SDBM, etc? Using the DBM Modules	Installing and Using Perl Modules Laziness, Impatience, and Hubris CPAN Using Modules Installing a Perl Module Unpacking the Module Source The Configuration Step The Guild Step The Build Step The Install Step Using CPAN.pm Using Module Documentation	Introduction to DBI/DBD in Perl The Old Way - DBPerls A Better Way - DBI/DBD Database Programming Handles Connecting to the Database Creating a SQL Query Getting the Results Updating Database Data Transaction Management Finishing Up
Module 10 : Perl SQL Programming	Module 11 : Perl/Tk	Module 12 : Extending Perl

DBI/DBD SQL Programming Error Checking in DBI Getting Connected Drivers Using Parameterized Statements Statement Handle Attributes Other Handle Attributes Column Binding The do Method BLOBs and LONGs and Such Installing DBI Drivers	Creating a Perl/Tk Application GUI Programming Overview Adding Widgets Scrolled Widgets Configuring Widgets Menus Using FileSelect Tk::Error and Tk::ErrorDialog Configuring Widgets Geometry Management Geometry Management with grid() The Frame Widget Defining Widget Callbacks Bindings Nonblocking I/O with fileevent()	Extending Perl with C/C++ Extending the Perl Interpreter Overview of Perl5 XSUBs Get Started with h2xs Set up the Perl Wrapper Class Write the XS Code The XS File Write Some Test Code What Do You Want? Returning Values on the Stack A Walk Through an XSUB Arguments to XSUBs Other h2xs Options
Module 13 : Embedded Perl	Module 14 : Module Development	
Embedding the Perl Interpreter Why Embed Perl? Embedding Perl in a C Program Compiling the Program perlmain.c Perl Data Types Macros and Functions Manipulating Scalars Memory Management Script Space Evaluating Perl Expressions Dynamic Loading Multiple Perl Interpreters	Module Development and Distribution Distributing Modules Get Started with h2xs Files Created by h2xs The Build Library (blib) Directory Unit Testing and test.pl Versions Using blib POD POD Translators Cutting a Distribution Other Niceties Makefile.PL	

# PRG660: Course Flask Web Development

Code: PRG660

#### Duration: 3 days

Price: € 1650

#### Audience Course Flask Web Development

The Course **Flask** Web Development is intended for developers who want to learn how to create Python Web Applications with the Flask Framework.

#### Prerequisites training Flask Web Development

To participate in this course prior knowledge of and experience with programming in **Python** is required.

#### **Realization course Flask Web Development**

The theory is treated on the basis of presentations. Illustrative demos are used to clarify the concepts discussed. There is ample opportunity to practice and theory and practice are interchanged. The course times are from 9.30 am to 16.30 pm.

### **Certification course Flask Web Development**

After successful completion of the course participants receive an official certificate Flask Web Development.

#### **Flask Web Development**

In the course Flask Web Development participants learn to develop Python Web Applications with the Flask Framework. The Flask Framework differs from other

frameworks because developers have a great deal of freedom in the components they use and are not tied to pre-cooked standard solutions. If developers want to use their own database engine or another method to authenticate users, this can be done without problems. Flask comes with a robust basic functionality but is designed to be expanded. Many extensions are available, but you can also use your own. The course starts with an overview of the features of the Framework, the initialization and the application structure. Next Flask templates with variables and control structures are discussed. Web Forms with View functions, Form rendering and Form Validation are also part of the course program. An overview is given of the possible database connections from a Flask application and attention is paid to both SQL and NoSQL databases. The implementation of Rest Services with Flask with the Flask Restless extension is also considered and ReSTful APIs for database models in for example SQLAIchemy are generated. Finally the testing of Flask applications and a number of popular Flask extensions are discussed.

Module 1 : Flask Intro	Module 2 : Templates	Module 3 : Web Forms
What is Flask? Flask Features Virtual Environment Serving HTML Initialization Application Structure Debug Mode Routes and View Functions Request Context Request Context Request Dispatching Request Hooks Responses	Rendering Templates Jinja2 Templates Variables Variable Rules Control Structures Templates in Files Templates Inheritance Filters and Links Custom Error Pages Static Files Localization Flask Moment	Form Classes Form Handling View Functions Form Rendering Security with CSRF Redirects Sessions Validation Flask WTF WTF Forms Flask Cookies Message Flashing
Module 4 : Data Access	Module 5 : Rest Services	Module 6 : Testing
SQL of NoSQL SQLAIchemy and MongoDB Model Definitions Relationships Database Operations Create and Insert Modify and Delete Querying Rows Shell Integration Database Migrations Flask Migrate	Resources and Id's URL Patterns Request Methods Request Bodies Response Bodies Pattern Parameters Flask Restless Versioning Content Negotiation Token Based Authentication Serializing Resources	Unit Testing Assertions Behavior Testing Flask Testing Life Server Exceptions JSON Handle Code Coverage Profiling Flask Test Client Testing Web Services

#### Module 7 : Extensions

Authentication
Flask-HTTPAuth
Flask Principal
Flask Login
Flask-OpenID
Flask-OAuth
Flask-Admin
Flask-Babel
Flask-Bcrypt
Flask Mail
Flask Sijax



# **PRG700: Course Regular Expressions**

Code: PRG700

#### Duration: 1 day

Price: € 550

#### **Audience Regular Expressions Course**

The course Regular Expressions is intended for application managers, developers and other interested parties who want to learn how regular expressions can be used for pattern matching in applications and tools.

#### **Prerequisites Course Regular Expressions**

To participate in this course general basic knowledge of computer systems and software applications is required. Programming experience is beneficial for a proper understanding.

### **Realization Training Regular Expressions**

The theory is treated on the basis of presentation slides. Illustrative demos are used to clarify the discussed concepts. The theory is interspersed with exercises. The course material is in English.

### **Certification Course Regular Expressions**

Participants receive an official certificate Regular Expressions after successful completion of the course.

#### **Course Regular Expressions**

The course Regular Expressions gives an overview of the capabilities of Regular Expressions and the way they work. After introducing the usages of Regular Expressions and the inner operation of Regular Expression engines, the syntax of Regular Expressions is treated. Attention is paid to the various meta characters such as those for quantification and choice and the escape sequences for special characters. Character classes are also discussed, including character ranges, characters to match at the beginning and end of a searched string and how to match on word boundaries. Here the difference between greedy and non-greedy Regular Expressions is addressed. Further advanced topics like the use of subgroups and backreferences are part of the course program. In this respect it is explained how subgroup expressions are defined by parentheses and how backreferences, addressed by an index or a name, ensure repeated execution of the Regular Expression. Finally it is discussed how regular expressions can be used for substitutions.

Module 1 : Intro Regular Expressions	Module 2 : Meta Characters and Quantifiers	Module 3 : Character Classes
What are Regular Expressions? Usages of Regular Expressions Origins of Regular Expressions Regex Mini Language Regular Expression Engines Simple Regular Expressions Inner Workings RegEx Engines Multiple Matches Case Sensitivity All Characters Count Non printable Characters Regex References Regex Tools	Meta Characters Match Any Character with dot Matching Multiple Dots Matching String Start and End Word Boundaries Alternatives with Pipes Matching Alternatives Quantifiers Optional Items Greediness Match Length Escaping Special Characters Common Character Escapes	Using Character Classes Shorthand Character Classes Negated Character Classes Start of Line versus Exclude Character Ranges Negating Characters and Ranges Meta Characters in Character Classes Matching Word Characters Matching Non-Word Characters Matching White and Non-White Space Matching Digits and Non-Digits Repeating Character Classes Named Character Classes
Module 4 : Subgroups and Backreferences	Module 5 : Substitutions and Modifiers	
Matching with Backreferences Match Character Next to Itself Using One Parentheses Group Multiple Parentheses Groups Turning of Backreferences Forward References Subgroups in Languages Named Subgroups Named Backreferences Lookahead and Lookbehind	Subsitution Example Substitution in Rx Toolkit Substitutions with s/// Substitute Operator More Substitution Examples Modifiers Perl Style Global Modifier Case Sensitivity Modifiers Replacement Patterns Transformations	





# Regular Expressions





# PRG800: Course Groovy Programming

Code: PRG800

Duration: 3 days

### Audience Groovy Programming Course

The course Groovy Programming is intended for Java developers who want to learn programming using the scripting language Groovy.

#### Prerequisites Course Groovy Programming

To participate in this course knowledge of and experience with programming in Java is required.

### **Realization Training Groovy Programming**

The theory is discussed on the basis of presentation slides and is interspersed with exercises. Illustrative demos provide further clarification of the discussed concepts. The course material is in English.

### **Certification Groovy Programming**

Participants receive an official certificate Groovy Programming after successful completion of the course.

#### **Course Groovy Programming**

In the course Groovy Programming participants learn programming in the Groovy scripting language that is sometimes referred to as Java++. Groovy is a JVM language

which means that Groovy scripts are interoperable with Java code and can therefore be used easily with Java in the same application. Groovy offers a natural extension to the Java syntax and its language facilities include simple data type syntax for lists, ranges, maps and regular expressions. After an introduction to the basic features of Groovy and installation of Groovy, it is discussed how to program with closures and builders. Attention is also paid to meta-programming in Groovy. Groovies support for dynamic typing, namely duck-typing, is discussed as well. Many more issues are on the course schedule and participants will find that they can quickly become productive with programming in Groovy.

Module 1 : Groovy Intro	Module 2 : Language Syntax	Module 3 : Data Structures
What is Groovy? Groovy Features HelloWorld in Java HelloWorld in Groovy Removing Noise Removing Boilerplate Dynamic Types Variable Interpolation POJOs on Steroids Script Support Differences with Java Running Groovy Scripts Groovy Development Kit	Groovy Variables and Data Types Wrapper Types Number Methods Groovy Strings Groovy Conditionals Switch Statement Groovy Operators Other Operators Elvis Operator Range Operator Groovy Loops For in Loop Each Loop	Groovy Lists List Methods List Manipulation Groovy Maps Map Methods Map Methods Map Manipulation Java Collection Interface Concrete Collections List Interface ArrayList Class LinkedList Class Set and SortedSet Map Interface
Module 4 : Methods and Closures	Module 5 : File I/O	Module 6 : Classes and Traits
Groovy Methods Method Parameters Return Values Locals versus Globals What are Closures? Benefits of Closures Groovy Closures Closure Parameters Closure Variables Closures in Methods Closures and Collections Find and FindAll Any and Every Collect	IO Basics Java Stream Classes Stream Types Nested Streams Groovy IO Reading Files Writing Text Data Conversion Streams DataStreams Java DataStreams Groovy Copy and Delete Directory Methods Traversing File Trees Executing External Processes	Groovy Classes Normal Class Duck Typing Property Support Inner Classes Inner Class Usage Anonymous Inner Class Abstract Classes Interfaces Constructors Fields Comparator Traits Meaning of this
Module 7 : XML and JSON Handling	Module 8 : Meta Programming	

XML Processing Options DOM Tree SAX Callbacks XML Support in Groovy MarkupBuilder JSON Object Representation JSON Array Representation JSON with Array and Objects Groovy Json Classes Producing JSON from Map Produce JSON From Pogo JSON with JSONSlurper

Groovy is Dynamic **Dynamic Methods** Adding Properties Meta Object Protocol Invoker Operation invokeMethod GroovyObject interface Groovy Interceptable Metaclass Intercepting Methods Expando GPath





# PRG800: Course Kotlin Programming

Code: PRG800

Duration: 2 days

#### Audience Course Kotlin Programming

The course Kotlin Programming is intended for developers who want to learn how Kotlin differentiates itself from Java and who want to learn how to program in Kotlin.

#### Prerequisites training Kotlin Programming

To participate in this course experience with programming is required. Knowledge of programming in Java is beneficial for the understanding.

#### **Realization course Kotlin Programming**

The theory is treated on the basis of presentations. Illustrative demos are used to clarify the concepts. There is ample opportunity to practice and theory and exercises are interchanged. The course times are from 9.30 to 16.30.

### **Certification course Kotlin Programming**

Participants receive an official certificate Kotlin Programming after successful completion of the course.

#### **Course Kotlin Programming**

In the course Kotlin Programming participants learn how to use the object oriented

and functional language Kotlin for software development. Kotlin is fully compatible with Java and the JVM version of its default library depends on the Java Class Library. The syntax of Kotlin is however much more concise thanks to type

simplifications in the Kotlin is mainly focused on the JVM, but also compiles to JavaScript or native code. The course Kotlin Programming deals with the many syntax simplifications in the Kotlin language and the differences with Java. Attention is paid to mutable and immutable variable declarations, extension methods, destructuring declarations, nested functions and the unpacking of parameters with the spread operator. With respect to object oriented programming, open abstract classes, public and final classes and primary constructor syntax are discussed. Finally, attention is paid to the use of Kotlin in scripts, in an interactive shell and as a procedural language.

Module 1 : Kotlin Intro	Module 2 : Language Syntax	Module 3 : Classes and Objects
What is Kotlin?	Numbers and Characters	Base class Any
Object Oriented Language	Booleans and Strings	Nested and Inner Classes
Functional Language	Arrays	Anonymous Inner Classes
Kotlin Characteristics	Destructuring Declarations	Type Aliases
Runtime and Performance	Immutable Collections	Primary Constructors
Running on JVM	Mutable Collections	Final is default
Byte Code Generation	Ranges	Open to Inherit
Interoperability	if else	Interfaces
Run as ECMAScript	when is	Visibility Control
Kotlin Architecture	for Loop	private and protected
Installing Kotling	while and do while	internal and public
Hello Kotlin Application	return, break, continue	Exception Handling
Module 4 : Extensions	Module 5 : Data and Sealed Classes	Module 6 : Generics
Function Extensions	data Keyword	Type Parameters
Functions outside main	Separate basic Data	Higher order Variables
Static resolvement	Data Holder Class	Abstracting over Types
fun keyword	Restricted Functionality	Generics Library
Add methods	Built-in toString()	Benefits of Generics
Remove methods	Built-in hashCode()	Interpretation Kotlin Compiler
Object Extensions	Extending Data Classes	out Keyword
Object in Factory method	Sealed Classes	When to use out
Difference with static In Java	Restricted Class Hierarchy	in Keyword
Extension property	Predefined Types	When to use in
static functionality	sealed Keyword	Class versus Type
companion object	Subclasses in same File	Generic Syntax

#### Module 7 : Functions and Delegation

Function Definition fun Keyword Lambda Functions Inline Functions Delegation Design Pattern by Keyword Property Delegation Using Lazy() Property as Input Return Type as Lazy Delegation.Observable()



# **PRG850: Course Grails Programming**

Code: PRG850

#### Duration: 3 days

**Price:** € 1400

### **Audience Grails Programming Course**

This course is intended for Java developers who want learn how to work with the new state of the art framework Grails.

#### **Prerequisites Course Grails Programming**

Knowledge of Java and the script language Groovy are required to participate in the course.

#### **Realization Training Grails Programming**

The theory is discussed on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

# **Certification Grails Programming**

Participants receive an official certificate Grails Programming after successful completion of the course.

### **Course Grails Programming**

The course Grails Programming teaches how to use the Grails framework for the pragmatic development of a Java EE application. The framework is based on the

Model Driven Design paradigm. In this Grails course participants will gain the knowledge to become a Grails developer. First, the most important foundations of the Groovy programming language, needed to become comfortable with Grails development, are treated. After this introduction, attention is entirely focused on the Grails framework. The full stack (model, view and controller) is explained in detail. Then all the important issues for the successfull development of a Grails application are on the agenda such as Services, Plugins and GORM.

Module 1 : Grails Intro	Module 2 : Grails Components	Module 3 : Grails Integration
The Essence of Grails Getting Started with Grails Understanding Domain Classes Understanding Controllers Understanding Views	Mapping URLs Internationalization Ajax Creating Web Flows GORM - Grails Object Relational Mapping Services	Integrating Grails Plugins Security Web Services Leveraging Spring Legacy Integration with Hibernate To Production: Grails and Tomcat





# Grails





# PRG850: Course Go Programming

Code: PRG850

#### Duration: 3 days

### Audience Course Go Programming

The course Go Programming is intended for developers who want to learn how to program in the Go language and who want to examine its capabilities.

#### Prerequisites training Go Programming

To participate in this course prior knowledge of and experience with programming in a modern programming language such as Java, C# or Python is necessary.

#### **Realization course Go Programming**

The theory is treated on the basis of presentations. Illustrative demos are used to clarify the concepts. There is ample opportunity to practice and theory and exercises are interchanged. The course times are from 9.30 to 16.30.

# **Certification course Go Programming**

Participants receive an official certificate Go Programming after successful completion of the course.

#### **Course Go Programming**

In the course Go Programming participants learn the syntax of the programming language Go and the style in which Go software is written. The Go programming

language is based on experiences gained with compiled languages such as **C**, C++ and Java and with interpreted languages such as Python and Ruby. Go is syntactically comparable to C, but with the additional advantages of memory safety, garbage collection, structural typing and concurrency based on Communicating Sequential Processes (CSP). The course Go Programming deals with the syntax of Go with data types and type inference, arrays, control flow and operators as well as functions and closures. Attention is also paid to the use of pointers in Go, the difference with pointers in C, pointer arithmetic and pointers to pointers. In terms of classes, the two ways in which Go offers an alternative to traditional inheritance are discussed. The first is embedding that can be considered as an automated form of composition or delegation. The second are the Go interfaces, which offer runtime polymorphism. The specific way of error handling in Go with the error and panic interface is also treated. Finally concurrency in Go with Go routines, sending and receiving with channels and worker pools is discussed.

Module 1 : Intro Go	Module 2 : Language Syntax	Module 3 : Operators and Control Flow
Origins of Go	Identifiers	Operator Types
Features of Go	Data Types	Miscellaneous Operators
Compilation Model	Integers and Floats	Operator Precedence
Type Inference	Strings and Booleans	if and else
Concurrency Support	Derived Types	Nested if
Go Routines	Variable Declaration	switch Statement
Native Binaries	Static Type Declaration	select Statement
Intentionally Exclusions	Type Inference	for Loop
Inheritance	Ivalues and rvalues	Nested Loops
Operator Overloading	Constants	Infinite Loops
Installing Go	String Literals	range Keyword
Comments	UTF-8 Sequences	break and continue
Go Programs	Immutability	goto Statement
Module 4 : Functions	Module 5 : Arrays	Module 6 : Pointers
Function Definition	Array Types	Address Operator
Function Declaration	Declaring Arrays	Pointer Type
Calling Functions	Initializing Arrays	Accessing Pointers
Local and Global Variables	Accessing Array Elements	Pointer Aritmetic
Parameters	Multi Dimensional Arrays	Comparion C Pointers
Return Values	Passing Arrays	Uage of Pointers
Call by Value and by Reference	Variables and Adresses	Dereferencing Pointers
Functions as Values	Slices	Nil Pointers
Function Closures	Nil Slice	Array of Pointers
Methods	Subslicing	Pointer to Pointer
Variadic Functions	Ien() and cap() Functions	Pointers as Parameters
Recursion	append() and copy() Functions	Type Casting
Module 7 : Data Structures	Module 8 : Error Handling	Module 9 : Concurrency
User Defined Data Types	Error is Type	Concurrency versus Parallelism
type Statement	Error Interface	Goroutines versus Threads
struct Keyword	Panic Interface	Multiplexing
Structure Definition	Reasons for Panic	Channels
Accessing Structure Members	Comparison to Exceptions	Race Conditions and Deadlock
Passing Structures	Using Multiple Return Values	Multiple Goroutines
Pointers to Structures	Unrecoverable Error	Declaring Channels
Object Oriented Programming	Programmer Error	Sending and Receiving with Channels
Structs versus Classes	Defer Execution	Blocking by Default
Composition over Inheritance	Recover Interface	Unidirectional Channels
Polymorphism	Runtime panics	Buffered Channels
Maps, Keys and Values	Goroutines	Worker Pools
make() and delete() Function	Stack Trace	WaitGroup
Interfaces	Analyzing Stack Traces	Select and Mutex





# **PRG900: Course Scala Programming**

Code: PRG900

#### Duration: 3 days

### Audience Scala Programming Course

This course is intended for Java, C# and other developers who want to learn programming in Scala or who want to explore the possibilities of Scala.

### Prerequisites Course Scala Programming

To participate in this course knowledge and experience with an object oriented programming language such as Java or C# is required.

#### **Realization Training Scala Programming**

The theory is discussed on the basis of presentation slides. Demos are used to clarify the concepts. The theory is interspersed with exercises. The course material is in English and a modern IDE is used.

# **Certification Scala Programming**

Participants receive an official certificate Scala Programming after successful completion of the course.

#### **Course Scala Programming**

In the course Scala Programming the syntax and capabilities of the Scala programming language are discussed. Scala combines the power of object oriented

Scala Scala



# Scala Programming





and functional programming and makes it possible to deliver similar functionality with substantially less source code as Java or C#. Attention is paid to the data types, variables, control structures and packages of Scala, the opportunities to expand Scala and the usage of Scala Frameworks like the Lift Web Framework and the Play Framework. Special attention is then given to the functional aspects of Scala such as first-class functions, higher order functions and the different ways of parameter passing in Scala. Other typical elements of the Scala language such as functions as objects, closures and deferred execution are discussed as well. Next attention is paid to collection types like lists, sets, maps and tuples and their higher order functions. Also the object oriented aspects of Scala such as inheritance, constructors, companion objects and overriding are treated and the support of duck typing is highlighted. The concept of Scala Traits and the uses of mixins is explained in detail. A separate modules is devoted to how Scala handles XML and JSON data. Finally attention is paid to concurrency in Scala and the usage of Akka Framework with Actors and Mail Boxes and asynchronous communication.

Module 1 : Scala Intro	Module 2 : Language Syntax	Module 3 : Functions and Closures
Introducing Scala Basic Syntax Scala Concepts Semicolons Scala Keywords Scala Characteristics Comments in Scala Interactive Shell Compiling Scala Scala HelloWorld Variables and Constants Java versus Scala Scala versus Java Scala Resources	Variables Variable Scopes Scala Data Types Scala Type Hierarchy If and Else Statements Multiple Selection For Loop For Yield Filtering While Loops Breakable Blocks Formatted Strings Scala Arrays Processing Arrays Multidimensional Arrays Ranges Scala Operators Reserved Symbols	Scala Functions Void Functions Call by Value Call by Name Named Parameters Variable Arguments Default Parameters Recursive Functions Nested Functions Methods versus Functions Anonymous Functions First Class Functions Higher Order Functions Partially Applied Functions Currying Functions Simple Closure
Module 4 : Collections	Module 5 : Classes and Traits	Module 6 : Pattern Matching
Scala Collections Scala Collection Hierarchy Scala Lists Nills and Cons Head and Tail List Concatenation List Fill and Tabulate Scala Sets Concatenation Min and Max Intersections and Unions Scala Maps Maps Concatenation Keys and Values Scala Tuples Tuple Access Scala Options GetOrElse	Scala Object Orientation Scala Classes Preventing State Change Object Singleton Constructors Visibility of Constructor Fields Companion Objects Inheritance Abstract Classes Traits Partial Trait Implementation Abstract Members Mixins Implicit Classes Scala JavaBeans	Unit returning Expressions Dealing with Any Match Expression Example Match Expressions Matching using case Classes Match with Variables Match with Sequences Extended Wildcard _* Scala versus Java Pattern Matching in Assignments Match with Types Matching on Tuples Pattern Matching with Option
Module 7 : Data Access	Module 8 : Scala Concurrency	

Unit returning Expressions Dealing with Any Match Expression Example Match Expressions Matching using case Classes Match with Variables Match with Sequences Extended Wildcard \_\* Scala versus Java Pattern Matching in Assignments Match with Types Matching on Tuples Pattern Matching with Option Concurrent Programming Akka Framework Actors in Akka Hello Akka World ActorSystem Actor Information Supervision Supervision Strategies Top Level Supervisors OneForOneStrategy Location Transparency Akka Guidelines

# **PRG950: Course Lift Programming**

Code: PRG950

Duration: 2 days

# Audience Lift Programming Course

This course is targeted at Scala and Java programmers who want to learn the Lift Framework.

### Prerequisites Course Lift Programming

Experience with and knowledge of Scala and Java is required to participate in this course.

### **Realization Training Lift Programming**

The subjects are discussed on the basis of presentation slides and demos. The theory is interspersed with exercises. The code is tested in different browsers. The course material is in English. The course times are from 9.30 up and to 16.30.

# **Certification Lift Programming**

Participants receive an official certificate Lift Programming after successful completion of the course.

#### **Course Lift Programming**

In the course Lift Programming the Lift Framework is treated. Lift is the most powerful, most secure web framework available today. The seven things that distinguish Lift

from other web frameworks are discussed in this course. First the security of Lift applications is discussed. Attention is paid how Lift applications are resistant to common vulnerabilities including many of the OWASP Top 10. Next the developer centric characteristics of Lift applications like, fast to build, concise and easy to maintain, are discussed. Also the designer friendlyness of Lift is treated showing how Lift applications can be developed in a designer friendly way. Furthermore the performance and scalability of Lift applications is part of the subject matter showing that Lift applications have high performance and scale in the real world to handle high traffic levels. Also the modularity of Lift applications is explained discussing the easy to integrate, pre built modules. And finally the interactivity of Lift applications is treated showing Lift's Comet support and Lift's ajax support.

Module 1 : Intro Lift	Module 2 : Lift Fundamentals	Module 3 : Forms in Lift
Implementing MVC with Lift Leverencing the Scala Language Defining the Model Creating a Template Writing Snippets Sprinkling Ajax Spice	Entering Lift Bootstrapping in Lift Using Lift Rules Resolving Classes Rendering with Templates Rendering with Views Lift Tags Merging HTML URL Rewriting Custom Dispatch Functions HTTP Redirects Cookies and Sessions	Checkbox Hidden Link Text and password Textarea Submit Multiselect Radio Select SelectObj untrustedSelect Uploading Files
Module 4 : Sitemap	Module 5 : Mapper and Record	Module 6 : Advanced Lift Architecture
Defining Site Map Creating the Link Class Using extLink Creating Menu Entries Using Nested Menus Setting the Global SiteMap Customizing the Display Using Hidden LocParam Controlling Menu Text Template LocParam LocSnippets	Adding mapper Constructing Mapper Enables Class Saving a Mapper instance Querying for Entities Control QueryPrograms Friendly Joins	Request/Response Cycle Exploring LiftResponse Managing Sessions ResourceServer Authentication
Module 7 : Lift and JavaScript	Module 8 : Ajax and Comet	
Using jQuery XML and javaScript Using JSON jQSHtml Object	Comet in Lift Understanding Actors Comet Actors Ajax in Lift Ajax Generators	





# Lift Programming



# PRG990: Course Scripting with Powershell

Code: PRG990

#### Duration: 3 days

Price: € 1499

### Audience Scripting with Powershell Course

This course is intended for Windows system administrators and other interested persons who want to use PowerShell to manage Windows and Windows applications.

#### **Prerequisites Course Scripting with Powershell**

To participate in this course general skills in working with computers and working knowledge of Windows is required.

### **Realization Training Scripting with Powershell**

The theory is treated on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises.

### **Course Scripting with Powershell**

In the course Scripting with PowerShell, participants learn to use PowerShell to manage Windows Servers. Incidentally PowerShell can also be used well for the management of other products and that too is discussed. Participants learn to use PowerShell from the command line and they learn to use the Integrated Scripting Editor. Further the operators, variables, data types and control flow structures are discussed. Also the calling of functions and creating of functions and the handling of errors is on the agenda. Accessing files, directories and registry with PowerShell is part of the course as well. Separate modules in the course are devoted to PowerShell





# Scripting with Powershell





Remoting and dealing with WMI. Finally attention is paid to some advanced topics such as performing administrative tasks related to Active Directory and IIS. After completing the course participants can use simple commands and ready made scripts to perform common management tasks faster.

Module 1 : Powershell Intro	Module 2 : Variables	Module 3 : Control flow
Basics of PowerShell Interactive shell Customizing PowerShell Aliases in PowerShell Basic commands PowerShell Providers dotNet in PowerShell Overview CmdLets Integrated Scripting Editor Commands and expressions	Variables Cmdlets Booleans Strings Numbers Arrays. Hash Tables. Built in variables XML Operators	Control Flow Branching Conditional Statements if, then, else Comparison Operators PowerShell Looping Script Blocks Variable Scope Object Properties
Module 4 : Functies	Module 5 : Remoting	Module 6 : WMI
Code reuse Using Functions Standard functions Creating functions Passing parameters Using return values Custom help Handling errors Accessing Files	Understanding WinRM and WSMan WSMan Security Enabling and Configuring Disabling Remoting The WSMAN Provider Trusted Hosts Configuring Remoting Using PSSessions	What is WMI? WMI Support Get-WMI Object Using the Get-WmiObject Cmdlet WMI Object Properties Instance Methods WMI Type Accelerators Namespaces and Classes Accessing WMI Performance Counters WMI Security
Module 7 : Powershell Advanced		
Registry Access Pipelines Active Directory		

Group policies IIS Exchange Other products

# PYT700: Course Python NumPy

#### Code: PYT700

Duration: 1 day

### Audience Python NumPy Course

The course Python NumPy is intended for scientists and Big Data analysts who want to use Python with NumPy and MatPlotLib for data analysis and data processing.

### Prerequisites Course Python NumPy

To participate in this course prior knowledge of Python programming is necessary. Knowledge of numerical methods is beneficial for the understanding.

#### **Realization Training Python NumPy**

The theory is dealt with on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

# **Certification Python NumPy**

The participants receive an official certificate Numerical Python after succesful completion of the course.

#### **Course Python NumPy**

In the course Python NumPy the Python packages NumPy en MatplotLib are discussed. These Python add-on libraries are very usefull for the creation of data

analysis and data processing applications. The course starts with an overview of NumPy and its sister library SciPy and how we can install these libraries. Next the NumPy's ndarray object and its methods are discussed. Attention is paid to many different array manipulation techniques. These methods process large data sets very efficiently. Next matrix handling with Numpy is treated and attention is paid to special routines for ordening, searching and comparing data in matrices. Finally the MatplotLib library is discussed. This library is closely integrated with NumPy and SciPy and this makes it a very powerful tool to create and plot complex figures. The course uses real world examples to visualize of one- and two dimensional data.

Module 1 : Numpy Intro	Module 2 : Common Functions	Module 3 : Matrices
What is NumPy? What is SciPy? Installing NumPy NumPy array object Selecting elements NumPy numerical types Data type objects dtype constructors dtype attributes Onedimensional slicing and indexing Multidimensional slicing and indexing Array comparisons any(),all(), slicing, reshape() Manipulating array shapes Stacking and Splitting arrays Converting arrays	Methods of ndarray Clipping arrays Compressing arrays Views versus copies ravel(),flatten(),transpose() Missing values Handling NaNs nanmean(), nanvar() and nanstd() File I/O Loading from CSV files mean() function Value range Dates Correlation Smoothing full() and full_like() functions	Working with Matrices ufuncs Creating matrices Universal functions Arithmetic functions Modulo operation Fibonacci numbers Bitwise functions Comparison functions Fancy indexing at() method Inverting matrices Finding eigenvalues Singular value decomposition Pseudo inverse Determinants
Module 4 : Special Routines	Module 5 : Plotting with MathplotLib	
Sorting partition() function Complex numbers Searching Array elements extraction Assert functions Almost equal arrays Equal arrays Ordering arrays Object comparison String comparison Floating point comparisons Unit tests	Simple plots Plot format string Subplots Histograms Logarithmic plots Scatter plots Fill between Legend and annotations Threedimensional plots Contour Plots Transformations Animation Projections	







# **PYT800: Course Scientific Python**

Code: PYT800

Duration: 2 days

### **Audience Scientific Python Course**

Scientists, mathematicians, engineers and others who want to use the SciPy Python library to create applications and perform data analysis.

#### **Prerequisites Course Scientific Python**

Knowledge of Python programming and the NumPy library is required. Some knowledge of numerical methods in scientific computing is beneficial for the understanding.

#### **Realization Training Scientific Python**

The theory is dealt with on the basis of presentation slides. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

### **Certification Course Scientific Python**

The participants get well after completion of the course, an official certificate Scientific Python.

#### **Scientific Python Course**

In this course the participants will learn what can be done with the Python SciPy

library for scientific computing. The course starts with an overview of the role of matrices to solve problems in scientific computing. Next the course proceeds by reviewing basic manipulation and operations on them, followed by factorizations, solutions of matrix equations, and the computation of eigenvalues and eigenvectors. Also interpolation and approximation is treated where advanced techniques are shown to approximate functions and their applications in scientific computing. Differentiation techniques to produce derivatives of functions are discussed as well as integration techniques showing how to compute areas and volumes effectively. The module Computational Geometry takes a tour of the most significant algorithms in this branch of computer science. And finally the course pays attention to statistical inference, machine learning, and data mining.

Module 1 : SciPy Intro	Module 2 : Matrix Calculations	Module 3 : Nonlinear Equations
What is SciPy Installing SciPy stack Anaconda distribution Constructing matrices Using ndarray class Using matrix class Sparse matrices Linear operators Scalar multiplication Matrix addition Matrix multiplication Traces and determinants Transposes and inverses	Singular value decomposition Matrix equations Least squares Spectral decomposition Interpolations Univariate interpolation Nearest-neighbors interpolation Other interpolations Least squares approximation Differentiation and Integration Numerical differentiation Symbolic differentiation Symbolic integration Numerical integration	Non-linear equations and systems Iterative methods Bracketing methods Secant methods Brent method Simple iterative solvers The Broyden method Powell's hybrid solver Large-scale solvers Optimization Unconstrained optimization Constrained optimization Stochastic methods
Module 4 : Computational Geometry	Module 5 : Descriptive Statistics	Module 6 : Inference and Data Analysis
Plane geometry Static problems Convex hulls Voronoi diagrams Triangulations Shortest paths Geometric query problems Point location Nearest neighbors Range searching Dynamic problems Bézier curves	Probability Symbolic setting Numerical setting Data exploration Picturing distributions Bar plots Pie charts Histograms Time plots Scatterplots and correlation Regression Analysis of the time series	Statistical inference Estimation of parameters Frequentist approach Bayesian approach Likelihood approach Interval estimation Frequentist approach Bayesian approach Likelihood approach Data mining Machine learning Trees and Naive Bayes Gaussian mixture models

Module 7 : Mathematical Imaging



# **Scientific Python**

Digital images Binary Gray-scale Color Alpha channels Smoothing filters Multivariate calculus Statistical filters Fourier analysis Wavelet decompositions Image compression Image editing Rescale and resize Swirl Image restoration Noise reduction

# SCR700: Course Linux Shell Scripting

Code: SCR700

Duration: 2 days

# Audience Linux Shell Scripting Course

The course Linux Shell Scripting is intended for administrators, developers and testers who want to learn how to write and understand shell scripts in a  $\underline{\text{Linux}}$  environment.

### **Prerequisites Course Linux Shell Scripting**

To participate in this course general knowledge of computers and operating systems is required. <u>Programming</u> experience is beneficial for a good understanding.

# **Realization Training Linux Shell Scripting**

The theory is treated on the basis of presentations. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

# **Certification course Linux Shell Scripting**

The participants receive an official certificate Linux Shell Scripting after successful completion of the course.

#### **Course Linux Shell Scripting**

In the course Linux Shell Scripting, participants learn to write shell scripts in a Linux





# Linux Shell Scripting

operations such as file manipulation, program execution and printing of text. In the course various shells such as the Bourne, C, Korn and Bash shell are discussed as well as the differences between these shells. Central topic is the Bash shell. Participants learn to write scripts with script commands, special characters, regular expressions and I/O redirection. Furthermore the sed batch editor, with which the editing of a text file can be handled automatically and the awk report generator are discussed. Participants will also learn how to process user input in scripts with command line parameters, options and redirection. Control flow and reading and writing of files are discussed. Finally calling and writing functions, passing parameters and receiving return values are on the course program. Module 1 : Shell Scripting Intro

environment. A shell script is a computer program that is executed by a command line interpreter. System administrators often use shell scripts for typical

Module 1: Shell Scripting Intro	Module 2 : Executing Commands	Module 5 : Bash Environment
What are shell scripts?	Fork and Exec	Shell Initialization Files
Types of shells	Built-in Commands	/etc/profile and /etc/bashrc
Bourne Shell	Creating Script Files	User Configuration Files
C Shell	Shell Syntax	/.bash_profile, /.bash_login and /.bash_logout
Korn Shell	Shell Functions	Global Variables and Local Variables
Bash shell	Shell Parameters	Exporting Variables
Invocations	Shell Expansions	Reserved Variables
Bash Startup Files	Redirections	Special Parameters
Interactive Shells	Init Script	Shell Expansion
Conditionals and Arrays	Writing Scripts	Command Substitution
Shell Arithmetic	Executing Scripts	Aliases
Directory Stack	Debugging Scripts	Bash Options
Module 4 : Regular Expressions	Module 5 : sed and awk	Module 6 : Conditionals
What is Grep?	What is sed?	Selection Statements
Grep and Regular Expressions	sed Commands	if-then-else Statement
Pattern Matching	Interactive Editing	Checking Files
Meta Characters	Deleting Lines	Checking shell Options
Repetition Operators	Range of Lines	Testing exit Status
Line Anchors	Non-Interactive Editing	String Comparisons
Word Anchors	Using sed in Scripts	Nesting ifs
Single Character Match	What is awk?	Boolean Operations
Wildcards	Print Selected Fields	The test Command
Character Ranges	Formatting Fields	Compound Condition Testing
Range Expressions	Special Patterns	case Command
Character Classes	awk Scripts and Variables	Initscript with case
Module 7 : Interactive Scripts	Module 8 : Repetitive Tasks	Module 9 : Functions
Displaying Messages	Iteration Statements	Function Syntax
echo Built-in	for Statement	Function Parameters
Escape Sequences	while Statement	Positional Parameters
read Built-in	until Statement	return Built-in
Prompt for User Input	Loop control	Variables in Functions
File Descriptors	Output Redirection	Array Variables and Functions
Redirection of Errors	Input Redirection	Functions on Command Line
File I/O	Break and Continue	Catching Signals
Closing File Descriptors	select Built-in	Signals with kill
Here Documents	shift Built-in	Handling traps

# SPR100: Course Scrum Fundamentals

Code: SPR100

#### Duration: 1 day

### Audience Scrum Fundamentals Course

This course is designed for employees who work in a Scrum project and others who want to experience what Scrum is.

#### **Prerequisites Course Scrum Fundamentals**

To join this course no specific skills are required. General knowledge of system development is useful for a proper understanding.

#### **Realization Training Scrum Fundamentals**

The theory is discussed on the basis of presentation slides. The theory is clarified using example projects from practice. On the basis of short case studies group discussions are performed which are guided by the trainer. The course times are from 9.30 up and to 16.30.

### **Certification Scrum Fundamentals**

Participants receive an official certificate Scrum Fundamentals after successful completion of the course.

#### **Course Scrum Fundamentals**

The essences of the Scrum project management method are discussed in this course.

In Scrum cooperation is vital for the success of a system development project. In contrast to traditional project management methods in the Scrum approach changes are taken into account and are responded to. Using Scrum an information system is fully built in a number of small iterations, designated as sprints. The various roles in a project such as Product Owner, Scrum Master, the Development Team and Manager in the background, are discussed. Next attention is paid to the so-called ceremonies such as the planning and the daily meeting. Also artifacts such as stories, the product backlog, priorities, estimates of the size and product timeline are addressed. Also the determination and execution of the sprints and the importance of the daily meeting are discussed. Finally attention is paid to to the advantages of the Scrum approach, but also the pitfalls of the method are discussed. The participants go through the entire Scrum process in just one day and will be able to identify the steps in a Scrum process.

Module 1 : Scrum Intro	Module 2 : Scrum Framework	Module 3 : Artifacts
What is Scrum? Project Management Team as unit Self-organized teams Business sets priorities Target accomplishment	Roles Product owner Scrum Master Development Team Manager Ceremonies Planning Daily meeting Review	Artifacts Stories Product Backlog Prioritization Size Estimation Product Timeline Sprints Life in the Sprints Sprint Backlog Sprint Planning Charts

#### Module 4 : Daily Practice

Daily meeting Todo tasks Completed tasks Obstacles Burn Down Chart Retrospective Benefits and Pitfalls





# Scrum Fundamentals





# SPR200: Course Scrum Master

Code: SPR200

Duration: 2 days

# Audience Scrum Master Course

Developers, testers, software architects, project managers, team leaders and others who want to obtain intimate knowledge of the Scrum software process and the responsibilities of the Scrum Master role.

#### **Prerequisites Course Scrum Master**

To join this course no specific skills are needed. It is recommended gain knowledge about Scrum prior to the course by reading articles.

#### **Realization Training Scrum Master**

The theory is discussed on the basis of presentation slides. The theory is clarified further by using example projects from practice. Exercises are performed on the basis of short practical agile examples. The course times are from 9.30 up and to 16.30.

#### **Certification Scrum Master**

Participants receive an official certificate Scrum Master after successful completion of the course.

#### **Course Scrum Master**

Scaling the Sprint Meetings Distributed Scrum Teams Infrastucture practices

In this course participants gain in two days the knowledge needed to guide Scrum

projects in the right direction. The implementation of a Scrum software process can have far reaching consequences for organizations and you can come up against difficulties causing not all benefits of Scrum to show up. The role of the Scrum Master is of great importance for the success of a Scrum project and in this course you are trained to be a Scrum Master. The course covers all subject matter that is required to become certified as a Scrum Master. Even participants who are already active as a Scrum Master and who want to deepen their knowledge and want to certify, can benefit from this training. The course examines the principles of Scrum and is focused in particular on the role of the Scrum Master in relation to the team. Also the roles, tools and meetings of the Scrum process are examined in detail. After completing the training, participants may pass an examination of Scrum.org online. This allows you to get certified and become a Professional Scrum Master. The exam is not included in the training and is optional.

etete

Module 1 : Scrum Intro	Module 2 : Scrum Process	Module 3 : Scrum Teams
What is Scrum? Scrum origin Scrum principles Other Software processes Scrum versus Agile Scrum Software process	Scrum Framework Scrum meetings Changes with Scrum Scrum and Development Scrum and Management Total Cost of Ownership	Scrum teams Team cooperation Team communication Self organization Multidisciplinary Teams
Module 4 : Scrum Terminology	Module 5 : Scrum Planning	Module 6 : Scrum and Changes
Product Backlog Management and Refinement Item Readiness Item Estimation Product Increment Definition of Done Sprint Backlog Update Scope Burndown Charts	Scrum planning Informing customers Scaling Scrum to large projects Project estimations Time boxes Risk Management Risk Reporting Determine progress	Scrum and changes Roles Rules Artifacts Organization effects Maintenance TCO optimalization
Module 7 : Scaling Scrum		
Scaling Teams Scaling Product Owners Scaling the Product Backlog		



# Scrum Master

tetetet

# SPR300: Course Agile Development

Code: SPR300

#### Duration: 2 days

### Audience Agile Development Course

This course is intended for developers, testers, project leaders and managers who want to gain knowledge about the design of an Agile software development process.

#### **Prerequisites Course Agile Development**

To join this course no specific skills are required. General knowledge of system development is beneficial for a proper understanding.

#### **Realization Training Agile Development**

The theory is discussed on the basis of presentation slides. The steps in an Agile development process are practiced using short case studies. The course times are from 9.30 up and to 16.30.

# **Certification Agile Development**

Participants receive an official certificate Agile Development after successful completion of the course.

#### **Course Agile Development**

In this course participants will learn the concepts, methods and practical techniques of

Agile software development. First the history of Agile and the fundamentals of strategic planning are discussed. Next the course will go through the complete cycle of a typical Agile development project with story writing, execution and

delivery. Attention is paid to the roles and role identification in Agile and to working in teams and achieving consensus. Also the importance of User Stories and estimation and planning in Agile are discussed. Finally it is addressed how iterations in Agile are performed, what is the place of Daily Standups en how metrics can be applied to Agile projects. During the course participants will be working in teams and practice Agile through hands-on exercises designed reinforce the understanding of the ins and outs of Agile development. Participants will leave this course with the practical knowledge required to begin with Agile development.

Module 1 : Agile Intro	Module 2 : Agile Roles	Module 3 : Agile Teams
Agile introduction History of Agile The Agile Manifesto Agile Methods Method differences	Role Overview Leadership in Agile User Roles User Roles for Stories User Roles for Requirements User Role identification Using Personas	Self Organizing Teams Empowered Teams Team Dynamics Achieving Consensus Meeting Rules Team Charters Vision and Strategy Road Mapping Building the Product Backlog
Module 4 : User Stories	Module 5 : Estimation	Module 6 : Planning
User Story Parts Acceptation Criteria Why User Stories? Gathering Stories Requirements Gathering Writing User Stories Acceptation Criteria Prioritization Story Mapping	Intro Estimation Conventional Estimation problems Agile Estimations Feature Estimation Free Planning Velocity Estimation of Data with Velocity	Strategic Agile Planning Release Planning Process Schedule Based Release Planning Scope Based Release Planning Keeping the Release Plan Iterative Planning Defining Done Capacity Planning Detailed Planning Planning for Chaos
Module 7 : Iterations	Module 8 : Metrics	
Iteration Execution Reviews Three Themes of Iteration Execution Conducting Daily Standups Foul Team Smells	Sprint Demo'sMetrics and Reporting Cost of Metrics Traditional Metrics Problems Understanding Agile Metrics Retrospectives Effective Retrospectives	



# **Agile Development**


# SPR400: Course Requirements with Use Cases

Code: SPR400

Duration: 2 days

#### Audience Requirements with Use Cases Course

This course is intended for system analysts and developers who want to learn how the functional requirements of systems can be specified with Use Cases.

### **Prerequisites Course Requirements**

There are no specific prerequisites for this course. General knowledge of system development is desirable for a proper understanding.

#### **Realization Training Requirements with Use Cases**

The theory is presented in the form of presentation slides. Brief case studies are used to practice the techniques. Demo projects clarify the discussed concepts. The course material is in English. The course times are from 9.30 up and to 16.30.

### **Certification Course Requirements with Use Cases**

Participants receive an official certificate Requirements with Use Cases after successful completion of the course.

#### **Course Requirements with Use Cases**

In this course the focus is on Use Case modeling which is a widely used analytical technique for specifying the functional requirements of a software system and the

creation of a framework for test case development. After an introduction to requirements in general, where the different types of requirements are discussed, the specification of Use Cases is addressed. Not only the description of the Use Cases in a textual document will be covered, but also the visualization of the Use Cases in a Use Case diagram. Attention is paid to how Use Case diagrams can be of help in the communication with the stakeholders and to gain insight into the size, complexity and the needed interaction with the system under consideration. Subsequently, it is discussed how a Use Case can be accurately described in a number of steps on the basis of a Use Case template. Primarily the focus is on the success scenario but it is also discussed how to specify different key secondary scenarios. Next also more advanced techniques in Use Case Modeling are addressed such as the use of Activity Diagrams with guards, branching of Use Cases with if and the use of iterations. Finally, there is room for discussion of the structuring of Use Cases through reuse via includes, extends and generalizations. The course is completed with attention to Use Cases in the context of prototypes, the interfaces of the system and the test plan.

Module 1 : Requirements	Module 2 : Use Case Intro	Module 3 : Use Case Modeling
Understanding Requirements Vision Documents Requirement Analysis Activities Requirement Types Functional Requirements Non-Functional Requirements Requirements Determination Requirements Classification Requirements Specification Conflicting Requirements Requirements Risks The glossary	Use Case approach Identifying stakeholders Use Cases Actors Identifying Actors Primary Actor Secondary Actors Information Define System scope System Context Diagram System Use Case Diagram Brief Use Case Description	Use Case Modeling Identifying Use Cases Use Case Diagram Use Case Modeling Steps Drawing Use Cases Describing Use Cases High Level Use Cases Expanded Use Case Use Case Template Prioritizing Uses Cases Packaging Use Cases
Module 4 : Scenario's	Module 5 : Advanced Use Case Modeling	Module 6 : Interfaces and Tests
Main success scenario Describing the steps Best Practices use case descriptions Other Scenarios Different types of scenarios Alternate scenarios and flows Alternate flows and exceptions Alternate scenario description	Activity Diagramming Adding decisions Guards and notes Branching with If Alternative Paths Scenarios Structuring Use Case Model Generalizations include and extends	Usability requirements Prototyping Prototype documentation Interface requirements Interface specifications Screen functionality Interfaces in iteration plan Testing Use Case Test plan





# **TLG100: Course Git Fundamentals**

# Code: TLG100

Duration: 1 day

# Audience Course GIT Fundamentals

This course is intended for anyone who wants to learn how to use Git as a version control system.

#### **Prerequisites GIT Fundamentals**

No specific prior knowledge is required to participate in this course. Experience with version management or software development is beneficial to good understanding.

### **Realization Training GIT Fundamentals**

The theory is discussed on the basis of presentation slides and is interspersed with exercises. Demo projects clarify the concepts discussed. The course times are from 9.30 am to 16.30 pm.

# **Certification GIT Fundamentals**

After successful completion of the course, participants receive an official GIT Fundamentals certificate.

#### **Git Fundamentals**

In the course Git Fundamentals participants learn how to work with the Git version

control system. Git Configuration options such as system, user and repository

configuration are discussed. Basic Git concepts such as working directory, staging area and the different types of repositories are treated. Frequently used Git Commands are explained such as git init, add, status, commit, log, diff and checkout. Also discussed is how you can refer to commits by means of a hash or by giving them a specific tag with a git tag. The splitting of branches for testing features with git branch is on the program as well. And the merging of these branches with git merge is discussed. Conflicts may arise during the merging of branches and the options for conflict resolution are explained. Working with remote repositories and the commands for this are also on the course schedule such as git remote, git fetch, git pull and git push. The course concludes with an overview of the tools available for Git version management.

Module 1 : Git Intro	Module 2 : Git Commands	Module 3 : Branching
What is Git?	git init	Branching Workflows
Version Control Systems	git add	Creating Branches
Git Installation	git status	git branch
Git Configuration	git diff	Switching Branches
Git Workflow	git commit	HEAD
Working Directory	git log	Branch Management
Staging Area	git checkout	Branch Pointer
Repositories	git tag	Master Branch
Git Tooling	git revert	Feature Branch
Git Bash	git reset	Remote Branches
Git Gui	git rm	git stash
Shell Integration	Other Commands	Amending Commits
Module 4 : Merging	Module 5 : Remotes	Module 6 : Tooling
git merge	Working with Remotes	Git Repositories
Merge Strategies	Bare repositories	Public Repositories
Merge Conflicts	git clone	Securing with SSH
Fork Maintenance	Sharing and Updating	Github
Conflict Resolution	git remote	BitBucket
Conflict Visualization	Setting Origin	Gitlab
git mergetool	git fetch	Git in Open Source
Resetting Master Branch	git pull	Gui Clients
Detached Heads	Deleting Remote Branches	Smart Git
git rebase	git push	Github Desktop
Rebasing versus Merging	Push Errors	SourceTree



# **Git Fundamentals**

# **TLG300: Course GIT for Developers**

Code: TLG300

Duration: 2 days

# Audience GIT for Developers Course

This course is intended for developers who want to use Git and GitHub for distributed version control.

### **Prerequisites Course GIT for Developers**

To participate in this course knowledge of and experience with software development in a modern language such as Java, C++, C# or PHP is required.

### **Realization Training GIT for Developers**

The theory is discussed on the basis of the presentation slides and is interspersed with exercises. Demo projects are used to clarify the concepts. The course material is in English.

# **Course GIT for Developers**

In the course Git for developers participants learn to understand the concepts of distributed version control and to begin with the GitHub suite of tools effectively. Attention is paid to repositories that can be located both local and remote, how versions can be committed and possibly undone. It is further treated what Git branching is and which strategies you can use for branching and merging. The control of Git from the command line is also discussed. Furthermore attention is paid to typical GitHub workflows, to undoing errors and history manipulation in a distributed





environment. Finally some of the available tools are treated like Revision Selection, Interactive Staging, Rewriting History and Debugging with bisect.

Module 1 : Git Intro	Module 2 : Git Branching	Module 3 : Git on the Server
What is Git? Git and DVCS concepts GitHub platform Repository Repository on the web Desktop repositories Recording Changes to the Repository Viewing the Commit History Undoing Things Tagging Document versioning Local and remote repositories Distributed version control Configuration and customization	Branching and Merging Brance Management Branching Workflows Remote Branches Rebasing Branching strategies Merge strategies Conflict resolution Multiple remotes Fork maintenance Temporary branching solutions	The Protocols Getting Git on a Server Generating SSH Public Key Setting up the Server Public Access Hosted Git (GitHub) Working with Remotes Remote repository interaction Repository integration
Module 4 : Distributed Git	Module 5 : Git Tools	
Distributed Workflows Contributing to a Project Maintaining a Project Synchronization Collaboration patterns Collaboration workflows Project management Git Configuration Attributes and Hooks Local and remote synchronization Shortcuts and efficiencies	Revision Selection Interactive Staging History reordering History editing Rewriting History Debugging with bisect Submodules Subtree Merging Accidental commits Ignore patterns uses Aliases beyond-the-basics	

# **TLG400: Course Continuous Delivery**

Code: TLG400

Duration: 2 days

Price: € 1199

# Audience Course Continuous Delivery

The course Continuous Delivery is intended for developers, testers and administrators who are involved in software development and who want to implement continuous delivery.

### **Prerequisites Course Continuous Delivery**

General knowledge and familiarity with software development, programming, testing and deployment is required to participate in this course.

#### **Realization Training Continuous Delivery**

The theory is discussed on the basis of presentations. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

### **Certificate Course Continuous Delivery**

Participants receive an official certificate Continuous Delivery after successful completion of the course.

# **Course Continuous Delivery**

 RELEASE
 BUILD

 Continuous
 DEPLOY

 RELEASE
 BUILD

 Continuous
 Continuous



delivery

DEPLO

TEST

Module 1 : Intro Continuous Delivery	Module 2 : Docker Essentials	Module 3 : Docker Applications
What is Continuous Delivery? Traditional Delivery Process Development, QA and Operations Shortcomings Traditional Delivery Benefits of Continuous Delivery Fast Delivery and Feedback Cycle Low Risk Releases Automated Deployment Pipeline Continuous Integration Automated Acceptance Testing Configuration Management DevOps Culture	Virtualization and Containerization Disadvantages of Virtualization Benefits of Containers The Need for Docker Isolation and Portability Installing Docker Docker Architecture Docker Components Docker Client Docker Server Docker Daemon Docker REST API	Building Docker Images Docker Commit Dockerfile Environment Variables Running Docker Containers Docker Container States Docker Networking Running Services Container Networks Exposing Container Ports Automatic Port Assignment Using Docker Volumes
Module 4 : Jenkins Essentials	Module 5 : Continuous Integration Pipeline	Module 6 : Automated Acceptance Testing
What is Jenkins? Extensibility by Plugins Distributed Operations Jenkins Installation Installing on Docker Initial Configuration Jenkins Pipeline Master and Slaves Vertical and Horizontal Scaling Test and Production Instances Configuring Agents Permanent Docker Agents Jenkins Swarm Agents Custom Jenkins Images	What is a Pipeline? Pipeline Structure Multi Stage Application Sections, Directives and Steps Commit Pipeline Pushing to GitHub Compile Stage Unit Test Stage Jenkinsfile Code Quality Stages Code Coverage and CheckStyle Scheduled Builds Development Workflows Non Technical Requirements	Acceptance Testing Intro? Docker Registry Artifact Repository Docker Hub Private Docker Registry Domain Certificates Access Restriction Building Images Pushing and Pulling Images Acceptance Test in Pipeline Acceptance Testing Stage Docket Compose Scaling Services Running Acceptance Tests
Module 7 : Configuration Management	Module 8 : Continuous Delivery Pipeline	Module 9 : Clustering with Docker Swarm

### Module 10 : Advanced Continuous Delivery

Managing Database Changes Understanding Schema Updates Database Migrations Using Flyway SQL Migration Script Backwards Compatibility Changes Non-Backwards Compatibility Changes Adding Columns Changing Code Merging Data Dropping Columns Avoiding Shared Database Preparing Test Data Parallelizing Pipelines Shared Libraries Types of Environment Production Environment Staging Environment Test Environment Development Environments Non Functional Testing Performance Testing Load Testing Stress Testing Scalability Testing Security Testing Non Functional Challenges Application Versioning Complete Jenkins File Server Clustering Docker Swarm Intro Setting up a Swarm Adding Worker Nodes Deploying a Service Scaling a Service Publishing Ports Rolling Updates Draining Nodes Multiple Manager Nodes Scheduling Strategy Docker Stack Specifying docker-compose.yml Kubernetes Scaling Jenkins

# **TLG500: Course Docker Containers**

Code: TLG500

#### Duration: 2 days

# **Audience Course Docker Containers**

The course **Docker** Containers is intended for developers who want to use Docker containers for application development.

#### **Prerequisites Course Docker Containers**

To participate in the course Docker Containers experience with the development of applications and related matters is required.

#### **Realization Training Docker Containers**

The theory is explained on the basis of presentations. The concepts are illustrated with demos. The theory is interspersed with exercises. The course times are from 9.30 to 16.30.

# **Official Certificate Course Docker Containers**

Participants receive an official certificate Docker Containers after successful completion of the course.

#### **Course Docker Containers**

In the course Docker Containers participants learn how Docker containers can be used in application development. Docker is a container manager that creates and

starts containers with a specific runtime environment. Unlike virtual machines, Docker uses resource isolation so that several independent containers can run in an operating system instance. Such a runtime environment is called a Docker image and is created with the instructions in a Docker file. The containers can be created, started and stopped just like processes. Docker machines are local or remote computers with an IP address on which the Docker service is running. Typically they are present in the cloud such as AWS, Amazon Web Services. In the course Docker Containers participants learn to make and run Docker containers. The various Docker commands such as run, pull, push, build and search are discussed. Attention is also paid to hosting Docker containers, the difference between Base and Child images and the contents of a Docker file. Cloud Deployment on various cloud platforms such as AWS and Google App Engine is also on the program. Finally attention is paid to the use of Docker containers in a Microservices Architecture, the use of multiple containers next to each other and in a cluster in combination with the Kubernetes cluster orchestration system.

Module 1 : Docker Intro	Module 2 : Docker Commands	Module 3 : Docker Images
What is Docker? Packaging Applications Containers versus Virtual Machines Cloud Deployment Amazon Web Services Docker Hub Installing Docker Running Docker Docker Daemon Docker Hub and Registry Docker Images Docker Client	pull Command run Command Container ID's ps Command docker container prune docker rm docker search Hosting in registry Hosting WebApps Publishing Ports Detached Mode Snapshots	Base Images Child Images Official Images User Images onbuild Version Using a Dockerfile FROM Keyword EXPOSE Keyword CMD Keyword docker build Development Workflow
Module 4 : Cloud Deployment	Module 5 : Multi Container Environments	Module 6 : Docker Network
Docker on AWS Elastic Beanstalk docker push Heroku Google App Engine AWS Console cmd for EB Dockerrun.aws.json Upload and Deploy Configuration Monitoring and Alarms	Multiple Services Multiple Containers Decoupling Application Tiers Microservice Architecture Scalablity Adding Containers Tweaking Base Images Custom Dockerfiles ADD command package.json Preparing Images	Exposing IP Address Bridge Network Default Network docker network Command network create Command Automatic Service Discovery Docker Compose docker-compose.yml Docker Machine Docker Swarm Kubernetes



# **Docker Containers**

# **TLG700: Course Web Testing with Selenium**

Code: TLG700

#### Duration: 2 days

Price: € 999

# Audience Course Web Testing with Selenium

The course Web Testing with Selenium is intended for testers and developers who want to learn Selenium Webdriver for the automated testing of Web Applications.

### Prerequisites Web Testing with Selenium

To participate in this course experience with programming is not strictly necessary. Knowledge of programming is beneficial for the understanding and can be obtained by participating in our course <u>Java for Testers</u> course.

### **Realization Training Web Testing with Selenium**

The theory is discussed on the basis of presentation slides. Demos are used to illustrate the theory. There is ample opportunity to practice. The course material is in English.

### **Certification Web Testing with Selenium**

Participants receive an official certificate Web Testing with Selenium after successful completion of the course.

### **Course Web Testing with Selenium**

In the course Web Testing with Selenium participants learn to work with Selenium

WebDriver and Selenium IDE with which web applications can be automatically controlled in different browsers. Via scripts in programming languages such as Java, C #, Python and Ruby the real outcomes of certain actions can be compared with the expected outcomes and the results can be reported. The course starts with recording user interaction with the Selenium IDE. Since the original Selenium IDE in FireFox is no longer working this done with a Selenium IDE in Chrome or the Katalon recorder in Chrome. The recorded test is then translated to a JUnit test script in Java or another programming language that automatically carries out the test. Topics in the course are the localization of page elements with id, name, link, CSS and XPath. The most common WebDriver functions are explained and applied. Navigating and manipulating page elements is also discussed. And also Advanced Web Driver functions such as adjusting wait time, the Page Object Model, Selenium Grid and Cross Browser Testing are discussed. Finally the integration of Selenium and the Cucumber Framework will be discussed. With Cucumber User Scenarios can be defined that can be translated to a JUnit Test with Selenium WebDriver.

Module 1 : Selenium Intro	Module 2 : Locating Elements	Module 3 : WebDriver Actions
What is Selenium? Test Automation Selenium WebDriver Test Recorders Chrome Selenium IDE Katalon Recorder Selenium IDE Record Test Case Running Scripts Adding Checks Verification Commands Assertion Statements General Selenese Commands WaitFor Commands Store and Echo Commands IDE Limitations	Loading WebDriver Web Elements Finding Elements Locator Types Locating by ID and Name Locating by Link Text Locating by CSS Selector Tag, ID and Class Tag and Attribute Inner Text Locating by DOM DOM getElementsByID DOM getElementsByID DOM getElementsByName Dom Name and Index Locating by XPath XPath Syntax	WebDriver Functions Actions on WebElements Clicking Check Boxes Clicking Radio Buttons Getting Attributes Sending Keys State Testing Clearing Submitting Select Items Table Lookup Get CCS Get Location Actions Class Switching Windows Waiting for Loading
Module 4 : JUnit Testing	Module 5 : Advanced Web Driver	Module 6 : Selenium and Cucumber
What is JUnit? JUnit Integration Assert Statements Running JUnit Tests JUnit Classes System Under Test Fixtures Annotations Test Suites Suite in Suite Suite TestRunner Special Case Parameterized Tests Executing Parameterized Tests	WebDriverWait Implicit vs Explicit Wait ExpectedConditions Dealing with Alerts Synchronisation Strategies Cookies Page Object Model Page Objects private Methods Page Factory Loadable Components Selenium Grid Hub and Nodes Cross Browser Testing	Test Driven Development Behavior Driven Development What is Cucumber? User Stories Scenarios Feature Files Gherkin Language Given and When Keywords Then and And Keywords Background Cucumber Options Step Definitions Selenium Cucumber Integration Data Driven Testing



# Web Testing with Selenium





# **TLG700: Course Elastic Search Fundamentals**

Code: TLG700

Duration: 2 days

### Audience Elastic Search Fundamentals Course

This course is intended for anyone who wants to learn how to use Elastic Search.

### **Prerequisites Elastic Search Fundamentals**

No specific prior knowledge is required to participate in this course. Knowledge of Java Development and the Spring Framework is beneficial for the understanding.

#### **Realization Training Elastic Search Fundamentals**

The theory is discussed on the basis of presentation slides and is interspersed with exercises. Demo projects clarify the concepts discussed. The course times are from 9.30 to 16.30.

### **Certification Elastic Search Fundamentals**

Participants receive an official Elastic Search Fundamentals certificate after successful completion of the course.

#### **Elastic Search Fundamentals Course**

In the Elastic Search Fundamentals course participants learn to use the Apache Lucene based search engine Elasticsearch for Enterprise queries on different types of data. Elasticsearch has a RestFull API with which you can quickly retrieve search

results in JSON format. Elasticsearch however also supports other data formats such as XML. Attention is paid to the different ways in which Elasticsearch can be deployed such as standalone, in a cluster, embedded or as a container. Also discussed is the fundamental unit of an Elasticsearch document that can represent any type of relevant application data. It is discussed what indices and mappings are and which data types can be assigned to fields. Various Elasticsearch APIs are covered such as the Cluster APIs, the Indices APIs, the term vectors API and the bulk document API. Attention is also paid to the Query DSL with which various queries can be executed such as full text queries, term level queries and geo queries. And it is discussed how mutations can be performed with queries. Next attention is paid to how Elasticsearch can be controlled from a Java application with Spring and Hibernate. Both the Java client API and the Java Rest Client are discussed. The course concludes with an overview of the Elastic Stack with supporting applications and tools such as Kibana for visualization, Beats, Logstash and Testing Kit.

Module 1 : Intro Elastic Search	Module 2 : Mappings	Module 3 : Using API
What is ElasticSearch?	Documents	Check Clusterhealth
Search Types	Indices	Indexes API
Full Text Search	Index Settings	Store Catalog Settings
Apache Lucene	Mappings	Create Index API
Restfull API's	Meta Fields	Catalog Index
JSON Documents	Data Types	Immutablity
Running ElasticSearch	Advanced Mappings	Get Mapping Types
Run as Data Node	Dynamic Mapping	Analysis Process
Run as Ingest Node	Analyzers	Index States
Standalone Instance	Tokenizers	Open and Close
Clustering	Character Filters	Single Document Operations
Embedded	Indexing	Bulk Document API
As Container	Internationalization	Fetch Documents
Module 4 : Query DSL	Module 5 : Java Client API	Module 6 : Elastic Stack
Search API	Transport Client	Hibernate Search
Request Body Search	Using Maven	JPA Entities
Query Parameters	Spring Configuration	Spring Boot Magic
Match All Query	Asynchronous Handling	ELK Abbreviation
Full Text Query	Aggregations DSL	Logstash Pipeline
Term Level Queries	Java Rest Client	Kibana Visualization
Joining Queries	JsonPath Library	TCP Socket Plugin
Inner Hits	Using Indexing API	Logback Encoder
Geo Queries	Data Transfer Objects	elasticsearch-head
Mutations by Query	Using Testing Kit	elasticsearch-HQ
Explain API	Integration Tests	search-quard



# Elastic Search Fundamentals



# TST200: Course FitNesse Acceptance Testing

Code: TST200

# Duration: 1 day

**Price:** € 599

Module 1 : FitNesse Intro	Module 2 : FitNesse Projects	Module 3 : Configuring FitNesse
What is FitNesse? Acceptance Testing Black Box Testing Functional Testing Test Automation FitNesse Web Server Wiki Tables and Engine Testing Engines SLIM and FIT Language Bindings	Edit Front Page FitNesse Variables Page Properties Root Page Test Pages Setup Page Teardown Page Fixtures Test Scripts Test Suites	Decision Tables Setters and Getters Configuring FIT Column Fixture Row Fixture Action Fixture Configuring SLIM Query Table Scenario Table Data Types
Module 4 : Writing Test Cases	Module 5 : Running Test Cases	
System under Test Accessing Elements Scenarios Data-driven testing Verification points Standard Keywords Own keywords Reporting Fitnesse plugins	Run individually Command line options Run as suite Naming in suites Selective run Smoke test Execution order Names of tables Modular approach	

# TST300: Acceptance Testing with Cucumber

Code: TST300

Duration: 2 days

Module 1 : Cucumber Intro	Module 2 : Gherkin Keywords	Module 3 : Step Definitions
Test Driven Development Steps in TDD What is BDD? BDD's Evolution BDD Second generation User Stories Scenarios BDD Tools Cucumber Features Scenarios	What is Gherkin? Gherkin Syntax Feature Files Gherkin Keywords Feature Keyword Background Keyword Scenario Keyword Given and When Keyword Then and And Keyword But Keyword * Keyword	Step Definitions Step Definition File Step Template Automation Script Step Implementation Cucumber Options dryRun Option monochrome Option features Option glue Option format Option
Module 4 : Data Driven Testing	Module 5 : Cucumber Hooks	Module 6 : Cucumber Tags
Parameterization Scenario Outline Executing Examples Data Tables Raw Methods Maps in Data Tables Test Step Implementation Matching Steps Failed Steps	What are hooks? Scenario Hooks Before Hook After Hook Lambda Style Around Hook Step Hooks BeforeStep and AfterStep Tagged Hooks	What are Tags? Scenario Subset Scoping Hooks Tag Placement Tag Inheritance Tag Expressions Run Scenario Subset Ignoring Scenarios Tags for Documentation
Module 7 : Cucumber Expressions	Module 8 : JUnit Integration	
Comparison to Regular Expressions Parameter Types Built-in Parameter Types int and float word and string anonymous Custom Parameter Types Optional text Alternative Text Escaping	What is JUnit? JUnit Integration Assert Statements Assert Class Fixtures Annotations Test Suites Suite in Suite Suite TestRunner Parameterized Tests	

# **TST400: Course Performance Testing**

Code: TST400

Duration: 2 days

Module 1 : Intro Performance Testing	Module 2 : Configuring Tooling	Module 3 : Creating Test Plan
Performance Aspects Memory Issues Timing Issues Response Times User Experience Perceived Performance BenchMarking Types of Performance Tests Load Testing Stress Testing Performance testing	Using JMeter Running JMeter Setting Classpath Configuring JMeter Non-GUI mode Using Gatling Gatling Recorder Recording Scenarios Running Gatling Isolate Processes Configure Users	Designing the Test Defining Workload Simulated Transactions Refining measurements Response Times Resource Usage Web Test Plan Set HTTP Request Headers Add Cookie Support Add File Reporter Saving and Running Test Plan
Module 4 : Creating Scripts	Module 5 : Test Execution	
Configuring Tree Elements Thinning a Test Thread Group Controllers Generative Controllers Logic Controllers Listeners Timers Assertions Configuring Elements Verification Points	Acquire Data Validate Tests and Tools Prepare for Execution Execute the Tests Refining Measurements Response Times Resource Usage Data Collection Conclusions and Recommendations Tuning and Debugging Reporting Results	

# VBA100: Course VBA Word Programming

Code: VBA100

#### Duration: 2 days

#### Price: € 999

### Audience VBA Word Programming Course

This course is designed for persons who want to learn to write templates in Word and who want to direct the input of forms with VBA.

### Prerequisites Course VBA Word Programming

Good knowledge of templates in Word and how Word deals with text, paragraphs and sections is required to participate in this course.

### **Realization Training VBA Word Programming**

The theory is treated on the basis of presentation slides. Illustrative demos provide further clarification of the concepts. The theory is interspersed with practical exercises.

## **Course VBA Word Programming**

This course discusses how VBA, Visual Basic for Applications, can be used to write templates and macros in Word and thus to access Word programmatically. Participants learn what the Word objects are and how they can be accessed. The course focuses on those parts of VBA that are important to control Word. The course participants learn to make templates that use custom dialogs, dynamic menus and toolbars. Attention is paid to the underlying VBA code. Next the syntax of VBA language is discussed with statements, variables and control flow constructs. An

important part of the course involves the use of functions and procedures in Word and





# **VBA Word Programming**





the participants also learn how to write their own and call them from VBA code. The so called Word object model and its properties and functions that allow you to control Word programmatically is addressed. Furthermore attention is paid to how databases can be accessed from Word with the DAO, Data Access Objects, object model. And finally events and their handling is on the course schedule. After completing this course, participants are able to create and maintain templates that are fully focused on their own corporate identity.

Module 1 : VBA in Word	Module 2 : VBA Syntax	Module 3 : Control flow
Templates in Word Template types Macro's and templates Recording and playing macro's Visual Basic Editor Programming in VBA Entering code VBA Code Statements Comments Interaction with the user Messagebox and Inputbox	Variables Constants Declarations Data Types Modules Objects Stepwise execution Breakpoints Debugging Syntax errors Error handling	Selections If Statements Else Statements If Then Else Select case Iterations For Next Do Loop With end with Arrays in VBA
Module 4 : Functions	Module 5 : Word Object Model	Module 6 : Events
What are functions? Parameters Arguments Finding functions Defining Functions Forms Making a user interface Creating dialog windows Moving the cursor	Word objects Object model Documents Paragraphs Sections Bookmarks Range Object What is DAO? Data Access Objects Word Database Access	What are Events? Type of events Reaction on events Event handling Event functions Domain functions

# VBA200: Course VBA Excel Programming

Code: VBA200

# Duration: 2 days

**Price:** € 999

Module 1 : VBA Intro	Module 2 : VBA Syntax	Module 3 : Control flow
Macro's in Excel Recording and playing macro's Visual Basic Editor Programming in VBA Entering code VBA Code Statements Comments Interaction with the user Messagebox and Inputbox	Variables Constants Declarations Data Types Modules Objects Stepping through code Breakpoints Debugging Syntax errors Error handling	Selections If Statements Else Statements If Then Else Select case Iterations For Next Do Loop With end with Arrays in VBA
Module 4 : Functions	Module 5 : Excel Object Model	Module 6 : Events
Wat are functions? Parameters Arguments Finding functions Defining functions Using Excel functions Creating dialogs	Excel objects Object model Workbooks Worksheets Cells Rows and columns What is DAO? Data Access Objects Excel Database Access	What are Events? Event Types Reacting on events Event handling Event function Domain functions

# VBA300: Course VBA Access Programming

Code: VBA300

Duration: 2 days

Price: € 999

# Audience VBA Access Programming Course

This course is intended for anyone who wants to learn how to use VBA, Visual Basic for Applications, and the DAO object model in Access to add functionality to forms, reports and databases.

### Prerequisites Course VBA Access Programming

Knowledge of and experience with Access is required to participate in this course. Programming experience is beneficial to a proper understanding.

### **Realization Training VBA Access Programming**

The theory is discussed on the basis of presentation slides. Illustrative demos provide further clarification of the concepts. The theory is interspersed with practical exercises.

### **Course VBA Access Programming**

The course VBA Access discusses how VBA, Visual Basic for Applications, can be used to program in Access and how the DAO object model can be used to access Access databases. In particular attention will be paid to those parts of VBA that are important for controlling Access. Participants will learn how to use VBA to retrieve information from an Access database and how the data can be filtered and sorted. Moreover, is discussed how forms and reports can get more functionality with VBA. Also the so called DAO, Data Access Object, object model is addressed and the





# **VBA Access Programming**





properties and functions of DAO that can provide access to all kind of databases. DAO is also applicable external to Access. Finally it is discussed how databases can be accessed with the DAO object model from Word and Excel.

Module 1 : VBA Intro	Module 2 : VBA Syntax	Module 3 : Control flow
Visual Basic Editor Programming in VBA Entering code VBA Code Statements Comments Execution step by step Breakpoints Debugging	Variables Constants Operators Parameters Arguments Procedures Functions Modules Objects	Selections If Statements Else Statements If Then Else Select case Iterations For Next Do Loop Arrays in VBA Multidimensional arrays Dynamic array declarations
Module 4 : Events	Module 5 : Forms	Module 6 : DAO
What are Events Type of events Reactions on events Event handeling Event functions Domain functions MsgBox in VBA	Forms Connecting to data Lookup information Using autolookup queries Forms synchronizing Control elements and fields Using calculated query fields Access commands DoCmd in Access	What is DAO? Data Access Objects Workgroups Record sets Word Database Access Excel Database Access

# XML100: Course XML Introduction

Code: XML100

Duration: 1 day

# Audience XML Introduction Course

The course XML Introduction is targeted at project managers, system analysts, application developers and system administrators who want to learn the basic concepts of XML and the usages areas of XML.

### **Prerequisites Course XML Introduction**

To attend this course basic knowledge of Internet and HTML is required.

### **Realization Training XML Introduction**

The theory is treated on the basis of presentation slides. The theory is accompanied by demos from practice. After discussion of a module, there is the opportunity for practice.

# **Certification XML Introduction**

Participants receive an official certificate XML Introduction after successful completion of the course

#### **Course XML Introduction**

XML is a language for describing structured documents and data and provides the technological basis for data sharing, publishing and e-business. In this course,

participants will gain a thorough understanding of the concepts in XML. The concepts in addition to the applications of the meta language XML are discussed. Attention is paid to the syntax of XML documents and the difference between well-formed and valid XML documents. The Role of Document Type Definitions (DTDs) and XML Schemas (XSD files) is discussed and several XML-derived languages, XML vocabularies, are discussed. Further the possibilities for presenting XML documents are treated where the focus is on the use of eXtended Stylesheet and Transformation Language (XSLT). Finally various applications of XML in practice are discussed, including Data Exchange, Publishing and Electronic Commerce. Possible follow-up courses are <u>XSLT Stylesheet</u>, <u>XML Schema</u> and <u>XQuery</u>

Module 1 : XML Introduction	Module 2 : XML Syntax	Module 3 : XML Validity
What is XML?	XML Document Structure	Document Type Definition
XML versus HTML	Node Tree	DOCTYPE Declaration
Structured Documents	XML Prolog	Internal and External DTD
Roots of XML	XML Elements	Element Declaration in DTD
XML Standard	XML Attributes	Attribute Declaration in DTD
Markup Languages	Well Formed Documents	XML Schema as DTD Successor
XML as Meta Language	Valid Documents	DTD to XML-Schema Conversion
Benefits of XML	Processing Instructions	XML-Schema Vocabulary
XML Technologies	Entity References	XML Schema Namespace
Applications of XML	CDATA Sections	Referencing XML Schema's
XML Validation with XML Schema	Character References	Simple and Complex Types
XML Presentation with Stylesheets	Comments	XML Schema Data Types
XML Transport with Web Services	Namespaces	User Defined Data Types

#### Module 4 : XML Formatting

Separate Content and Presentation What is XSL? XSLT Stylesheets How does XSLT work? What is XPath? Stylesheet Blueprint Templates xsl:value-of xsl:for-each xsl:for-each xsl:fi Using Predicates Applying templates XSLT as Transformation Language



</course>
<

price currency="Euro">450</price>

<?xml version="1.0"?>| <courses> <course duration="1" id="a1"> <code>XML100</code> <title>XML Oveniew</title>

xml version="1.0"?>| courses> <course duration="1" id="a1"> <course duration="1" id="a1"> <course duration="3" id="a1"> <course duration="3" id="a3"> <course duration="3" id="a3"</course duration="3" id="a3"

</courses>

# XML150: Course XML Fundamentals

Code: XML150

#### Duration: 2 days

# Audience XML Fundamentals Course

The course XML Fundamentals is designed for project managers, system analysts, application developers and system administrators who want to learn the fundamentals of XML and how and where XML can be used.

### **Prerequisites Course XML Fundamentals**

To attend this course basic knowledge of the Internet and HTML is required. Knowledge of databases and object oriented programming is beneficial to the understanding of the subject matter.

### **Realization Training XML Fundamentals**

The theory is covered using presentation slides. Demos are used to clarify the theory. There is ample opportunity to practice.

### **Certification XML Fundamentals**

Participants receive an official certificate XML Fundamentals after successful completion of the course.

#### **Course XML Fundamentals**

In the course XML Fundamentals participants will gain a thorough understanding of

Variant Monpulation Technologies With Manual States Structure & Data Typing With States Structure & Data Typing Structure & Data Typi

# XML Fundamentals



the fundamentals of XML. The concepts of the meta language XML are discussed as well as the usage of XML. Attention is paid to the syntax of XML documents and to the difference between well-formed and valid XML documents. The role of Document Type Definitions (DTDs) and XML Schema's (XSD files) is discussed as well as the importance of several derived XML languages, so called XML vocabularies, is explained. Also the different ways to present XML documents is covered. In this respect the focus is on the use of eXtended Stylesheet and Transformation Language (XSLT) style sheets and the XPath language to locate and address certain parts in an XML document. The new XQuery XML standard for accessing XML and relational data stores is covered and the syntax and practical applications of XQuery are explained. Also attention is paid to Web services for exchanging data between heterogeneous distributed systems and related standards such as SOAP and WSDL. Finally the access of an XML document from a programming language with the DOM and SAX API is discussed. Possible follow-up courses are <u>XSLT Stylesheet</u>, <u>XML Schema</u> and <u>XQuery</u>.

Module 1 : XML Introduction	Module 2 : XML Syntax	Module 3 : XML Validity
What is XML?	XML Document Structure	Document Type Definition
XML versus HTML	Node Tree	DOCTYPE Declaration
Structured Documents	XML Prolog	Internal and External DTD
Roots of XML	XML Elements	Element Declaration in DTD
XML Standard	XML Attributes	Attribute Declaration in DTD
Markup Languages	Well Formed Documents	XML Schema as DTD Successor
XML as Meta Language	Valid Documents	DTD to XML-Schema Conversion
Benefits of XML	Processing Instructions	XML-Schema Vocabulary
XML Technologies	Entity References	XML Schema Namespace
Applications of XML	CDATA Sections	Referencing XML Schema's
XML Validation with XML Schema	Character References	Simple and Complex Types
XML Presentation with Stylesheets	Comments	XML Schema Data Types
XML Transport with Web Services	Namespaces	User Defined Data Types
Module 4 : XML Formatting	Module 5 : XQuery	Module 6 : XML Programming
Separate Content and Presentation	What is XQuery?	XML Processing Options
What is XSL?	XQuery Processing Model	The XML DOM
XSLT Stylesheets	Types of Queries	Building a DOM Tree
How does XSLT work?	Where is XQuery used?	Node Interface Model
What is XPath?	XQuery and other Technologies	XML Access through DOM
Stylesheet Blueprint	Basic Syntax Rules	Navigation and Manipulation
Templates	Selecting Nodes	Simple API for XML
xsl:value-of	XPath Expressions	SAX Callbacks
xsl:for-each	FLWOR Expressions	XML Parsing Models
xsl:if	Structure of an XQuery Module	Pull Parsing versus Push Parsing
Using Predicates	XQuery Functions	What is JAXB?
Applying templates	Element Constructors	JAXB Architecture
XSLT as Transformation Language	Conditional Expressions	JAXB Binding Life Cycle

Module 7 : XML Web Services

What is a Web Service? Role of Interface Interoperability Web Service Stack RPC Style Web Services Document Style Web Services What is SOAP? SOAP Structure SOAP Message SOAP Messages as Payload SOAP Header What is WSDL? Basic Structure WSDL WSDL and Code generation Service Orientation WS Standards Overview

# XML200: Course XSLT Stylesheets

Code: XML200

#### Duration: 3 days

## Audience Course XSLT Stylesheets

The course <u>XSLT</u> Stylesheets is intended for Application developers, Web developers and project managers who want to learn XSLT to apply XML formatting and transformation.

#### **Prerequisites Course XSLT Stylesheets**

Knowledge of <u>HTML</u> and <u>XML</u> syntax is required. Knowledge of the basic concepts of programming is not required but beneficial.

#### **Realization Training XSLT Stylesheets**

The theory is discussed on the basis of presentation slides. The theory is clarified using demos and interspersed with exercises. The course material is in English.

# **Certification XSLT Stylesheets**

Participants receive an official certificate XSLT Stylesheets after successful completion of the course.

#### **Course XSLT Stylesheets**

The course XSLT Stylesheets discusses how XSLT, eXtensible Stylesheet and Transformation Language can be used to format and transform XML documents. The

 

 XSL Transform (XSLT)
 XSL Transform (XSLT)

 Source Tree
 Result Tree (element and attribute nodes)

# **XSLT Stylesheets**



course is based on the latest version of XSLT, version 3, but the differences with previous versions are also highlighted. The course deals with transformations from XML to HTML, from XML to XML and from XML to other text formats such as plain text and JSON. Furthermore, attention is paid to navigating through XML documents via XPath expressions and the various search axes of XPath are explained. There is plenty of attention for the many functions that XPath and XSLT have to offer and that can be called during the transformation process. An important element of XLST is the principle of template matching. This therefore forms the central element of the course. The XSLT elements that have to do with flow control are discussed, as well as those for copying elements. Furthermore, the use of generic templates and the use of apply templates is discussed as well as named templates and their calling with call-template. The transformation of attributes to elements and vice versa is also on the course schedule. Finally, we look at transformations from XML to various other formats. In this respect, a separate module is devoted to the transformation language XSL-FO and transformations to PDF.

Module 1 : XSL Intro	Module 2 : XML to HTML Transformation	Module 3 : XPath
The XSL Family	Adding Stylesheets	XPath Expressions
What is XSLT?	Extracting Values from XML	XPath Node Types
What is XPath?	Navigating the XML Document	XPath Context
XSL's Position	Using Predicates	Location Path Syntax
XSLT Stylesheets	Iterating through XML Elements	XPath Axes
XSLT Versions	Creating HTML Tables	Predicates
StyleSheet Blueprint	Absolute and Relative Path	XPath Operators
The XSLT Vocabulary	Conditional Processing	XPath Functions
Template Rules	Accessing Document Parts	Node Tests in XPath
Default Templates	// Operator	XML Schema Data Types
XSLT Document Structure	Access XML Data with Nodelists	Sequences
xsl:output	Internal hyperlinking	For Expressions
XSLT Processors	Numbering	Conditional Expressions
Processors Implementations	Sorting	Namespace Nodes
Module 4 : XSLT Variables	Module 5 : XSLT and XPath Functions	Module 6 : XML to XML transformation
xsl:variable	Numeric Functions	Transformation Language
xsl:variable with Subtree	String Functions	xsl:element
Working of xsl:variable	Boolean Functions	Selecting all Elements
Variable Scope	Accessor Functions	Selection all Attributes
Global Variables	Node Functions	name function
Changing Variables	Sequence Functions	xsl:attribute
Tracing Output	Any Uri Functions	Copying Elements
xsl:param	Context Functions	Generalization
Passing Parameters to Templates	Duration and Time Functions	Multiple Applicable Rules
Named Templates	Error and Trace Functions	mode attribute
Calling Templates	QName Functions	xsl:include
Call by Reference	Regular Expression Functions	xsl:import
Module 7 : XML to Text Transformation	Module 8 : Advanced XSLT	Module 9 : XSL-FO
Text Formats	Using Typing	What is XSL-FO?
Plain Text Files	Strong Typing	XSL-FO Documents
CSV files	XML Schema Validation	Layout Master Set
Transforming JSON	Temporary Trees	Page Sequence Masters
Handling Entity References	User-Defined Functions	Page Template Layout
White Space Handling	Grouping by Value	XSL-FO Areas
xsl:space	Grouping in Sequence	XSL-FO Pages
Preserving space	Multiple Result Trees	Static Content
xsl:preserve-space	Output Serialization	Flows
Stripping Space	Character Mapping	Page Sequences
xsl:strip-space	Character Substitutions	Tree for Pagination

# XML300: Course XML Schema

Code: XML300

Duration: 3 days

# Audience Course XML Schema

The course XML Schema is designed for developers of XML data structures and XML applications and project managers who want to use XML schemas to validate the contents of XML documents.

#### Prerequisites Course XML Schema

To join this course knowledge of the basic syntax of XML is required.

#### **Realization Training XML Schema**

The theory is discussed on the basis of presentation slides. Demos are used to illustrate the theory. There is ample opportunity to practice. The course material is in English.

# **Certification Course XML Schema**

Participants receive an official certificate XML Schema after successful completion of the course.

#### **Course XML Schema**

In the course XML Schema you will gain extensive knowledge on the syntax and usage of the XML Schema language, as successor to Document Type Definitions

(DTDs). XML Schema is used to define XML vocabulaires that define the structure, element names and contents of XML documents. Several XML Schema design models are discussed like Russian Doll, Salami Slice and Venetian Blind. Attention is given to declaring simple and complex elements and types and imposing constraints on the content of elements. Also the creation of user defined simple data types using facets and regular expressions is among the subjects discussed. Namespacing is an important part in the XML Schema specification and is given a lot of attention. Also the creation of complex derived data types and the difference between derivation by restriction and derivation by extension are discussed. Further attention is paid to modularization of schemas and the usage of the include and import mechanisms. Finally some advanced topics like Open Content Models and the XML Schema handling of keys and references and the removing of redundancy by identity constraints are addressed.

Module 1 : XML-Schema Intro	Module 2 : XML Schema Basics	Module 3 : XML Schema Models
Why XML Schema? What is XML Schema? Markup Languages Well Formed and Valid Documents Document Type Definition (DTD) DTD Limitations XML Schema as DTD Successor XML Schema Features Typical Use of XML Schema Use of XML Schema Other Uses of XML Schema Schema Validators	XML Schema Components DTD to XML Schema Conversion DTD Vocabulary XML Schema Vocabulary Target Vocabulary Referencing XML Schema XMLSchema-instance Namespace Multiple Levels of Checking Element Cardinality Simple and Complex Types Simple Type Restricted to Integer Complex Type with Attribute	Declaration versus Definition Global versus Local Element Declarations Global and Local Declarations Referencing Global Declarations Anonymous and Named Types Three Design Approaches Salami Slice Design Russian Doll Design Venetian Blind Design Combined Design Design Comparisons
Module 4 : XML Schema Data Types	Module 5 : Derived Simple Types	Module 6 : XML Schema Documentation
XML Schema Data Types String Data Types Language Data Type Name Types ID Types Qualified Names and URI's Binary String Encoded Types Primitive Numeric Data Types Derived Numeric Data Types Boolean Data Type Date Data Types ur-type and anyType	Creating Simple Types Derived Numeric Simple Types Simple Types by Restriction Available Facets Enumerations and Patterns simple Type from simple Type Fixing Facet Values Regular Expressions Meta Characters Quantifiers and Character Classes List Type and simple Types Union type and simple Types	Annotating Schema's Annotation element Allowed Locations Annotation Location Inlining Annotation Documentation Element Appinfo Element Optional Attributes source Attribute xml:lang Attribute Defining Semantics
Module 7 : Namespaces	Module 8 : Complex Types	Module 9 : Derived Complex Types
Namespaces of XML Schema XML Schema Namespace TargetNamespace Referencing XML Schema Namespace Scope Default Namespace Symbol Spaces Name Conflicts What is in the Namespace? Namespace Qualification elementFormDefault attributeFormDefault Rules for using Namespaces	Simple Content and Attributes Local Attribute Declarations use Attribute Grouping Attributes Grouping Elements Global Group Definition Choice Element Fixed Element Values Default Element Values Sequence and Choice Any order with all Empty element nil and Mixed content	Derived Complex Types Deriving by Extension Deriving by Restriction Prohibiting Derivations Element Substitution Substitutable Elements International Clients substitution Group Features Substitution with Derived Types Blocking Element Substitution Transitive and Non-Symmetric Abstract Elements Abstract complexType
Module 10 : Schema Modules	Module 11 : Schema Extensions	Module 12 : Uniqueness and Keys



?xml version="1.0"?> xxd:schema xmlns:xxd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.courses.org" targetNamespac="http://www.courses.org" elementFormDefault="qualified">

course" maxOccurs="unbo

elementFormUetault="qualite (sd:element name="courses"> <xsd:complexType> <xsd:sequence> </xsd:sequence> </xsd:sequence> </xsd:complexType> ysd:alement>

</xsd:element>

ma> </xsd:sche

<?xml version="1.0"?>
<xxd schema xmlns xxd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.courses.org"
targetNamespace="http://www.courses.org"
elementFormDefault="qualified">
<xxd element name="courses"> 

# XML Schema

Schema Modularization Including Schema Documents Using include Chameleon Effect Namespace Coercion Redefining Types Using redefine Redefine no targetNamespace Importing Schema Documents Using import any Element Extension Element Instance with any Namespace Extension Elements anyAttribute Element Extension Attribute Instance with anyAttribute Namespace Extension Attributes Open Content Global and Local Openness Uniqueness and Keys Unique versus Key Key Requirements Combination Key Unique Unique Elements Key Referencing key Element keyref Element

# XML500: Course XQuery

Code: XML500

Duration: 3 days

# Audience XQuery Course

The course **XQuery** is designed for developers who wish to apply XQuery in practice for the selection of XML data.

### **Prerequisites Course XQuery**

To participate in this course basic knowledge of  $\underline{XML}$  and  $\underline{HTML}$  syntax is required. Experience with programming and Structured Query Language (SQL) is beneficial for a good understanding.

### **Realization Training XQuery**

The theory is discussed on the basis of presentation slides. Demos are used to clarify the theory. The theory is interspersed with hands-on exercises. Usage is made of modern XQuery tools and the course material is in English.

### **Course XQuery**

The XML language XQuery is central to the XQuery course. The participants learn how XQuery can be used to select and transform XML data. Attention is paid to the syntax of XQuery, the XQuery specification and the parts of XQuery. Also the relationships of XQuery with other XML vocabularies such as XPath, XSLT and XML Schema are discussed. Furthermore the data types, built-in functions of XQuery and the different XQuery expressions are discussed. The so called flower expressions



xquery version "1.0"; { for Scourse in doc("courses.xml")/courses/course where Scourse/@duration=1 order by Scourse/title return {data(Scourse/title)} }

# XQuery



xquery version "1.0"; { for \$course in doc("courses.xml")/courses/course where \$course/@duration=1 order by \$course/title return {data(\$course/title}}

which are characteristic for XQuery are treated in detail. Finally attention is paid to some more advanced applications of XQuery like joins, the use of XQuery in combination with other technologies and for the accessing of relational data.

Module 1 : XQuery Introduction	Module 2 : XQuery Syntax	Module 3 : Node Selection with XPath
What is XQuery? XQuery Motivation XML versus relational model Requirements Query Language for XML Three parts of XQuery XQuery Language Characteristics Types of Queries Physical Representations to Query Where is XQuery used? XQuery's position XQuery and other technologies XQuery Specifications	Basic Syntax Rules XQuery Functions Structure of an XQuery Module Example XQuery Module XQuery Expressions Example Books.xml Path Expressions Predicates Element Constructors Other Query Expressions FLWR Expressions Conditional Expressions XQuery Comparisons	XPath 2.0 Expressions XPath 2.0 Data Model XPath 2.0 Data Types XPath 2.0 Changes XPath 2.0 sequences XPath Context XPath Location Steps What is an Axis? Peer Axis Types Descendent Axis Types Location Path Syntax Predicates For expressions Quantified expressions Conditional expressions Functions
Module 4 : XQuery Data Types	Module 5 : Joins and Node Generation	Module 6 : Functions and operators
XPath 2.0 data Model Infoset and PSVI Three Building Blocks Items Atomic Types Atomic Values XQuery Type Hierarchy XML Schema Types String Types Date and Time Types Bumeric Types Binary Data Types	Expressing Joins Constructing Nodes FLWR Expressions For versus Let Element Constructors Attribute Constructors Text Constructors Other Constructors Sample Queries	Functions and Operators XPath 1.0 Functions XQuery 1.0 Functions Functions with Regular Expressions Enhanced String Functions Functions for Sequences XPath 1.0 Operators XQuery 1.0 Operators Comparision Operators Types Issues Constructor Functions User Defined Functions
Module 7 : Advanced Concepts		
String Search Queries using Namespaces Listing Namespaces Listing Target URI's Recursive parts Explosion Access to relational Data		

# XML505: Course XBRL Overview

Code: XML505

#### Duration: 1 day

# Audience XBRL Overview Course

The course <u>XBRL</u> Overview is intended for accountants and financial specialists who must learn to read and understand XBRL documents.

### **Prerequisites Course XBRL Overview**

To participate in the course XBRL Overview knowledge of the basic syntax of <u>XML</u> is required.

### **Realization Training XBRL Overview**

The theory is discussed on the basis of presentation slides. Demos are used to clarify the theory. There is ample opportunity for practical exercises. The course material is in English.

# **Certificate Course XBRL**

Participants receive an official certificate XBRL Overview after successful completion of the course.

#### **Course XBRL Overview**

The course XBRL Overview covers the basic concepts of the Extensible Business Reporting Language (XBRL). XBRL is an XML vocabulary that is used for the





# **XBRL Fundamentals**





exchange of financial data and the preparation of financial reports. The foundations of XBRL are based on XML standards such as XML Schema, XLink and XPointer and the essentials of these standards are discussed. The preparation of an XBRL taxonomy in the form of an XML Schema is also discussed. In this context we should actually speak of a Discoverable Taxonomy Set consisting of various XML Schemes and linkbases in the form of XML files with meta information. The course covers how draft reports can be converted into XML tags and how meta-information is assigned to these tags. The various linkbases for labeling, presenting and referring information are discussed. And also the calculation linkbase with which simple business rules can be enforced is treated. Attention is also paid to XBRL instance documents, the specification of numerical contexts and attributes and the representation of data in more dimensions. Finally some applications of XBRL will be discussed as well as a.o. the extensibility of XBRL with more complex business rules in the XML Formula linkbase.

Module 1 : XBLR Intro	Module 2 : Taxonomies	Module 3 : Linkbases
What is XBRL?	What are Taxonomies?	What are Linkbases?
What is XML?	XBRL Taxonomy Standards	Characteristics of Linkbases
XML versus XBRL	Taxonomy Parts	Roles and Arcroles
XLink and XPointer	Schematic Overview	Label Linkbase
Benefits of XBRL	Taxonomy Selection	Language Support
Taxonomy and Instance	Taxonomy Schema	Label Linkbase Roles
Extension Taxonomies	Concept Definitions	Reference Linkbase
XBRL Specifications	Item Data Types	Reference Linkbase Roles
Financial Reporting Overview	XML Schema Data Types	Calculation Linkbase
GAAP and IFRS	Taxonomy Elements	Calculation Relations Concerns
Legacy Reporting Workflow	Concept Attributes	Definition Linkbase
Single Source of Data	Taxonomy Relationships	Definition Linkbase
XBRL Business Report	Linkbases References	Reference Linkbase
Module 4 : Instance Documents	Module 5 : Dimensions	Module 6 : XBRL Modules
Instance Documents	Mapping Intro	Extending XBRL
XBRL Root	Multidimensional Model	Taxonomy Types
XBRL Root Attributes	Pivot Tables	XBRL Formula
Facts	Mapping Dimension Values	Formula Linkbase
Items and Tuples	Mapping Context	Using XPath
Contexts	Mapping Facts	Binding Variables
Context Entity	Mapping Footnotes	Value Expressions
Context Period	Complete Mapping	Calculations Results
Context Scenario	Options and Properties	Standard Predicates
Context Examples	Functions and Filters	Equality Testing
Units	Hypercubes	Inline XBRL
Editing Default Units	Types of Hypercubes	XBRL Versioning
Validation	Editing Hypercubes	XBRL Table Linkbase

# XML700: Course XSL-FO

# Code: XML700

Duration: 2 days

XSL

Regis Regis

XSL-FO

XSLT

# Audience XSL-FO Course

The course XSL-FO is intended for designers who want to learn about the formatting of XML and layout of documents with XSL-FO.

### Prerequisites Course XSL-FO

To participate in this course basic knowledge of XML and HTML syntax is required. Experience with programming is beneficial for a good understanding.

### **Realization Training XSL-FO**

The subjects are discussed on the basis of presentation slides. Demos are used to clarify the treated concepts. The theory is regularly interspersed with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

# **Certification XSL-FO**

Participants receive an official XSL-FO certificate after successful completion of the course.

# **Course XSL-FO**

This is a highly practical course for developers who need to use XSL Formatting

Objects to produce sophisticated screen or print output from XML source data. The role of XSLT to produce XSL-FO documents and the role of XSL-FO processors to produce other output like PDf is explained. Attention is paid to the layout master set and the different page master that can be chosen. Also the difference between static content and page flow content is discussed. The use of many elements and and attributes of the XSL-FO language is treated in detail. Both open source tools and XML Spy are used for the delivery of this course.

Module 1 : XSL-FO Intro	Module 2 : Blocks and Inlines	Module 3 : Page Layout
An XSL-Overview XPath, XSLT and XSL-FO The XSL-FO Processor Output Formats PDF, Postscript, RTF Short Form and Inheritance Structure of XSL-FO Document Flow Objects Layout Objects Auxiliary Objects Flows and Flow Mapping Formatting Object Content	Blocks and Inlines Formatting Blocks Box Model Border Space, Indent and Padding Background Formatting Text Inline Elements Inheritance wrapper Creating Lists Iist-block item, label and body	About Pagination Simple Pagination Complex Pagination Working with Page Sequences Working with Areas Area Types Area Components Area Positioning Block and Inline Elements Tables and Lists Working with Graphics and Color Character Properties and Fonts
Module 4 : Page Flow	Module 5 : Additional Content	
Page Sequence Masters Page Masters Viewport Area Cross Document Links Creating and Index Creating Table of Contents Working with Headers Page Layout and Flow Region Mapping Stylesheet Considerations	Links and Graphics External Links Internal Links Leaders, Markers and Page Numbers The Output Leaders Columns, Keeps, and Breaks Columns Keeps Breaks	