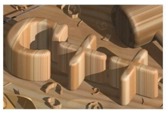
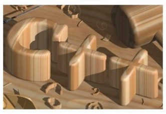
C++ Programming

Duration: 5 days **Price: € 2999**

Open Schedule



C++ Programming



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.

Content 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.

Differences C and C++

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.

References

Subsequently the new C++ reference variables are discussed. Attention is paid to both Lvalue and Rvalue references.

C++ Classes

An important element of the course is the C++ class concept and C++ implementation of object-oriented 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.

Inheritance and Polymorfisme

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.

Operator Overloading

C++ has the option to give existing operators a different meaning and this phenomenon is discussed in the module operator overloading.

Templates en Standard Template Library

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).

Exception Handling

Finally exception handling and how this is implemented in C++ is addressed. A follow up course for the course C++ Programming is Advanced C++ Programming.

Modules Course C++ Programming

Module 1 : Intro C++	Module 2 : Variables and Types	Module 3 : References
Intro C++ C++ TimeLine Comments in C++ Namespace std Output and Error Stream Standard Input Stream cin and Strings Formatted Output Variable Declaration Scope Resolution Operator Inline Functions Default Function Arguments Overloading Functions Range based for loop	Standard Types Type Inference Auto Keyword Deduction with decltype Initialization Null Pointer Constant Strongly Types Enums Variable Scope Namespaces Using keyword and Directive Block Usage User Defined Literals Storage Classes const Qualifier	References Reference Initialization References and Pointers Rvalues and Rvalues in C Rvalues and Rvalues in C++ Reference to Constant Passing References Comparison Parameter Passing References as Return Values Returning lvalue Returning Reference to Global Rvalue References Comparing Reference Types Rvalue Reference Usage
Module 4 : Classes	**Module 5 : Dynamic Memory Allocation**	**Module 6 : Inheritance**
Classes and Objects Classes in C++ Class Declaration Class Sections Constructor and Destructor Uniform Initialization Header and Sources Files Class Implementation Advantages Access Functions References to private Data this Pointer static Members Constant Objects Member Objects Friends	new and delete Operators Dynamic Arrays Classes with Pointer Data Assignment Operator Self-Assignment Problem Chained Assignments Assignment and Initialization Copy Constructors Passing Objects Returning Objects Passing References to Objects Move Constructor Move Assignment Operator Perfect Forwarding Delegating Constructors	Inheritance Derived Classes in C++ Class Hierarchy Redefining Member Functions Derived Class Constructors Base - Derived Class Conversion Pointer Conversions Virtual Functions Polymorphism Dynamic Binding Virtual Function Table Pure Virtual Functions Abstract Classes Multiple Inheritance Virtual Derivation
Module 7 : Operator Overloading	**Module 8 : Exception Handling**	**Module 9 : Templates**
Operator Overloading Overloading for Numeric Types Complex Type Example Overloading Rules Overloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Operators as Friend Unary Overloading Operator	Exception Handling in C++ Memory Exhaustion Handling Throwing Exceptions try Block catch Handlers Multiple catch Handlers Template Array Class Exceptions Array Class catch Order throw List	What are Templates? Template Functions Template Specialization Template Parameter List Class Templates Template Parameter Scope Template Function Statics Template Class Statics Inclusion Compilation Model Templates and Friends
Module 10 : STL		
Standard Template Library STL Core Components STL Library Components STL Containers Vector Container Deque Container List Container STL Iterators STL Algorithms STL Allocators		