

Go Programming

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.

Duration: 3 days**Price: € 1850**[Open Schedule](#)**Go Programming**

Content Course Go Programming

In the course Go Programming participants learn to develop applications using the concise and efficient Go programming language. Go is syntactically similar to C, but with the added benefits of memory safety, garbage collection and structural typing. The concurrency mechanisms in Go are based on Communicating Sequential Processes (CSP) and Go programs can get the best out of multicore machines. Go is a fast statically typed and compiled language but feels like a dynamically typed and interpreted language.

Go Intro

The course starts with a discussion of Go's syntax with data types, type inference, arrays, control flow and operators. The difference between rvalues and lvalues is also covered and attention is paid to immutable data.

Functions and Closures

The functions and parameter passing in Go is treated. The distinction between call by value and call by reference is explained. Variadic functions, recursion and closures are also discussed.

Pointers

Then it's time to pay attention to the use of pointers in Go. The difference with pointers in C is explained as well as pointer arithmetic, nil pointers and pointers to pointers.

Classes

Classes in Go are also part of program. The course explains the two ways in which Go offers an alternative to traditional inheritance. The first is embedding and can be seen as an automated form of composition or delegation. The second is the use of Go interfaces, which provide runtime polymorphism.

Error Handling

Also the specific way of error handling in Go with the error and panic interface is discussed. In this respect attention is also paid to the recover interface and the analysis of stack traces.

Concurrency

Finally concurrency in Go with Go routines, sending and receiving with channels and worker pools and synchronization mechanisms are covered.

Modules Course Go Programming

Module 1 : Intro Go	Module 2 : Language Syntax	Module 3 : Operators and Control Flow
Origins of Go Features of Go Compilation Model Type Inference Concurrency Support Go Routines Native Binaries Intentionally Exclusions Inheritance Operator Overloading Installing Go Comments Go Programs	Identifiers Data Types Integers and Floats Strings and Booleans Derived Types Variable Declaration Static Type Declaration Type Inference lvalues and rvalues Constants String Literals UTF-8 Sequences Immutability	Operator Types Miscellaneous Operators Operator Precedence if and else Nested if switch Statement select Statement for Loop Nested Loops Infinite Loops range Keyword break and continue goto Statement
Module 4 : Functions	Module 5 : Arrays	Module 6 : Pointers
Function Definition Function Declaration Calling Functions Local and Global Variables Parameters Return Values Call by Value and by Reference Functions as Values Function Closures Methods Variadic Functions Recursion	Array Types Declaring Arrays Initializing Arrays Accessing Array Elements Multi Dimensional Arrays Passing Arrays Variables and Adresses Slices Nil Slice Subslicing len() and cap() Functions append() and copy() Functions	Address Operator Pointer Type Accessing Pointers Pointer Arithmetic Comparison C Pointers Uage of Pointers Dereferencing Pointers Nil Pointers Array of Pointers Pointer to Pointer Pointers as Parameters Type Casting
Module 7 : Data Structures	Module 8 : Error Handling	Module 9 : Concurrency
User Defined Data Types type Statement struct Keyword Structure Definition Accessing Structure Members Passing Structures Pointers to Structures Object Oriented Programming Structs versus Classes Composition over Inheritance Polymorphism Maps, Keys and Values make() and delete() Function Interfaces	Error is Type Error Interface Panic Interface Reasons for Panic Comparison to Exceptions Using Multiple Return Values Unrecoverable Error Programmer Error Defer Execution Recover Interface Runtime panics Goroutines Stack Trace Analyzing Stack Traces	Concurrency versus Parallelism Goroutines versus Threads Multiplexing Channels Race Conditions and Deadlock Multiple Goroutines Declaring Channels Sending and Receiving with Channels Blocking by Default Unidirectional Channels Buffered Channels Worker Pools WaitGroup Select and Mutex