GO Lang Programming

Go (or Golang) is an open source programming language designed to build fast, reliable, and efficient software at scale.

Google uses Go specifically for its large networks of servers, and Go also powers much of Google's own cloud platform. Developers use Go in application development, web development, in operations and infrastructure teams, and much more. It is the language of Cloud Native infrastructure and software development.

As Go's popularity and adoption continue to increase, we'll see even greater usage and more creative implementations of the language!

Duration:

- 6 days

Detailed course outline:

Day 1

Go Overview and Hello World

- Go Introduction
- Go Versions
- Why Go?
- Installing Go
- Go Editors and IDEs
- Visual Studio and Go
- Hello World in Go
- Adding Initialization Behavior
- Accessing Command Line Arguments
- Go Programs Consist of
- Noteworthy Omissions from Go

Go Programming Core Elements

- Go Tokens
- Variables
- Assignments
- Short Variable Declarations

- Arithmetic Operators
- Constants
- Pointers
- Strings
- fmt.Printf() formatting

Flow of Control in Go

- Relational Operators
- If statement
- Using else and else if
- Nesting if statements
- Logical Operators
- Switch Statement
- expression switch
- type switch
- For loops
- Nested for Loops
- while style for loops
- Break, Continue and Goto

Day 2

Data Types

- Go Data Types
- Numbers
- Arrays
- Slices
- Ranges
- Maps
- List and Ring

Functions

- Defining Functions
- Zero, single and multiple parameter functions
- Returning values
- Variadic parameter lists
- Call by Reference
- Anonymous Functions
- Deferring a Function call
- Higher Order Functions
- Closure
- Recursion

Day 3

Structs: Object-based Programming

- Struct Introduction
- Defining a Struct
- Variables of type struct Person
- Initializing Structs
- Comparing Structs
- Structs are value types
- Passing Structs to Functions
- Defining Constructor Functions
- JSON and Structs
- Anonymous Structs
- Adding Behavior

Interfaces

- interface Introduction
- Defining an interface
- Creating User Defined Interface
- Implementing an Interface
- Using an Interface Type
- Working with interfaces
- Built-in Interfaces
- Stringer and Interface
- The empty interface: interface {}

Day 4

Error Handling

- Error Handling Introduction
- Basics of Error Handling
- Returning an Error
- Custom Error Types
- Handling Errors Idiom

Files and Paths

- File Information
- Checking if a file exists
- Obtaining a file reference
- File Permissions
- Reading Files
- Writing Data to Files

- Renaming and Deleting files
- Random Access Files
- Paths
- Directories

Day 5

Packages and Modules

- Packages
- Go Packages
- Installing Packages
- Importing a Package
- Defining Packages
- importing Custom Packages
- Installing Packages
- Package Initialization
- Modules

Testing

- Introduction to Go Testing
- Package testing
- Test Functions
- Parametric Tests
- Benchmark Tests
- Skipping Tests
- Test Setup / Shutdown behavior
- Verifying Examples

Day 6

Go and Relational Databases

- Examples of Relational Databases
- MySQL
- Connecting to an SQL Database
- Query a Database
- Insert into the Database
- Update a row in the Database
- Delete a row in the Database