

Course: Golang Programming Training

Duration: 5 days (8 hours per day)

Day 1

- Benefits of Modern Programming
- Comparing existing OOP language with Golang
- Benefits of Functional Programming
- Advantages and Disadvantages of most famous languages.
- Introduction to Golang
- Background, History
- Static Typing in Golang
- Building, Running a Go Program
- Environment Setup
- Testing Environment
- Basic data types, variables declaration
- Numerical and Boolean operations
- Different ways of variable handling
- For and if statements
- Go Switch
- Golang Arrays,
- Declaring and initializing an array
- Looping through an array
- Slicing an array
- Copying to an array
- Appending elements to an array
- String and String arrays
- String operations and formatting
- String methods Index, Contains, Count, Replace
- Splitting a string, Sorting an array of strings
- Introduction to maps
- Creating a map, adding members to the map
- Length of a map, adding new key/value pair
- Deleting a key in the map
- Introduction to functions
- Creating simple functions,
- function signatures and syntax
- Passing parameters/arguments to functions
- Returning single

- multiple values from functions
- Variadic functions and samples
- Functions as expressions in Golang
- Inline functions in Golang
- Writing recursive functions in Golang
- Deferring statements in a function
- Panic and Recover methods in Golang

Day 2

- Introduction to Structs
- Creating, declaring a struct and struct members
- Initializing a structure with initial values
- Assigning methods to structs
- Introduction to interface types
- Creating an interface
- Implementing an interface and achieving
- polymorphism
- Polymorphism demo
- Introduction to Pointers
- Pass by value and pass by pointers
- Getting the address of a variable
- Passing the address of the variable
- Dereferencing the pointer
- Using new function
- Using go's built-in packages (os, log, io/ioutil)
- Creating/opening a file
- Reading from a file, Writing to a file
- Closing a file, logging errors using log package
- Type conversions
- Converting int to float and vice-versa
- Strconv - string conversion functions ParseInt, ParseFloat
- Introduction to Go concurrency
- Converting sequential flow to concurrent flow
- Using Goroutines
- Introduction to channels
- Creating a channel using make function
- Passing/Receiving information thru channels
- Channel synchronization through sync package
- Using waitGroup from sync package

- Using mutexes for synchronization
- Concurrency patterns and their usage
- Fan-In, Fan-Out, Generator, Pipelines
- Context Implementation and Cancellation

Day 3

- Introduction to http programming in Golang
- Using net/http package
- Creating a web server
- Creating a route handler using http.HandleFunc method
- Installing third party packages using go get
- 3rd party frameworks (Gin Gonic)
- Setting up multiple routes and handlers
- Returning strings, json from the http methods
- Defining data models using structs
- Encoding/decoding structs to Json using Json Encoder/Decoder
- Serving static files
- Parsing request and url parameters
- Parsing request body
- CRUD operations
- Validating requests params/body
- Handling single/multiple file uploads
- Using middleware architecture
- Creating Protocol Buffers
- Creating .proto files
- Creating gRPC based services
- Creating gRPC client and server
- Introduction to Kafka
- Kafka message patterns
- Creating Producers and Consumers for Kafka

Day 4

- Introduction to database programming in go
- Built in sql package
- Installing mysql drivers for go
- Opening and closing sql connections
- Connection pools
- Importing and exporting data

- Sample tables creation
- One to many relationship
- Performing Inserts and Updates
- Querying single/multiple rows
- Populating results into structs
- Handling nested structs
- SQL Joins and struct handling
- Handling pagination
- Ordering and Filtering
- Handling database errors
- Using GORM package
- Packages for NoSQL databases
- Exposing MySQL data via REST api
- Go format, Go Build, Go run commands
- Compiling Go programs for various platforms
- Building go apps for deployment
- Develop Serverless application using Golang

Day 5

- Microservice fundamentals
- 12 Factor Application
- Host Components as Service
- Docker container Overview WRT Golang
- Creating MicroServices Applications
- Using Service Components
- Deployment Models
- Docker containers creation using Go Apis
- Different models of Deployments
- Serverless vs Kubernetes
- working with Serverless Applications
- Profiling and Mocking techniques
- Inbuilt methods
- Tuning using Design Patterns
- Coding Best Practices