GO Programming- 3 Day Workshop

Course Overview

This course introduces the GO programming language from Google and provides learners with an overview of Go's special features. Upon completing the course, learners will have gained the knowledge and skills needed to create concise, efficient, and clean applications using Go.

Pre-requisites

Participants who are familiar with any programming language can take this course

Lab Set-up

Windows, Linux or Mac

Golang installation, Postman,

Chrome

Course Outline

Day 01

Unit Title	Unit Outline
Why Go?	The Beginnings of Go
	Go vs Other languages
	Supported Platforms, Cross Compiling
	Key Distinguishing Features
Setting Up Go	Downloading and Installing Go Setting up Go
	Environment Variables
	Why do we need Git, Mercurial, etc.?
	Go Playground
Basic Program, Go Tools	Hello World, packages, import and main
	Go build
	Go run
Working with Strings	String Functions
	String Formatting
Variables and Assignment	var, := new
	Multiple assignment
	Values
	Variables
	Constants
Errors	Errors in Go, Struct Error, Errors.Is and Errors.As
	Error Conventions

	Custom Errors panic and recover
	defer
Functions	Writing a Function, Passing function as first class citizen
	Return Values
	Multiple Return Values
	Closures
Pointers, Parameters, Return Values	Pointers and problems with pointers
	Parameters
	Pass by Value, Pass by Reference
Arrays, Slices, Maps, for	For
	Arrays Slices
	Maps
	Range, continue, break, , fallthrough

Day 02

Unit Title	Unit Outline
OOP - Structs Encapsulation, Inheritance, Polymorphism	Structs
	Struct Members
	Anonymous Struct Members
	Methods on Structs
	Pointer and Value Receivables in struct methods
	How structs take place of objects
	Data Hiding
	Struct Composition
	Polymorphism
Interfaces,	Interfaces, Type assertion, Type switch, Interface pollution
Workspaces, go get, and managing dependencies	Workspace Directory Structure go get command
	go env variables and library search
	vendor Directory dep
Goroutines, Parallelism	Concurrency with goroutines
	Concurrency and Parallelism
Handling Race Conditions	Example of Race Condition
SyncGroup, Wait, Mutexes	Sync, Wait
	Mutexes
	Deadlocks with Mutexes
	RW Mutexes

Day 03

Unit Title	Unit Outline
Channels	Buffered channels
	Directional channels
	Channel types
	Select channels
	Project - Build a simple Get request engine
Concurrency in the real world	Rate limiting
	Worker pool Cancelling Goroutines using context Select Fan Out Pattern
Templates and Data formats	HTML and Text templates
	JSON
	JSON marshalling, JSON unmarshalling
Building web servers	Using regex in Go http package
	Running a Web Server and Handling Requests
	HTTP Return Codes
	Regex Routes
	Variables
	Serving Static Files
	Context gorilla package Installing gorilla mux

	Routing URLs
	Sub routers
Build REST services	What is REST?
	CRUD and REST
	HTTP Requests and REST
	A REST Project in Go