Golang Training Curriculum

Day wise Curriculum:

Schedule	**Lecture**	**Topics Covered**
Day 1	Getting started with GO	▪ Overview about the language ▪ Features of the language ▪ Why GO? ▪ GO programs ▪ Compilation and execution ▪ Local Environment Setup ▪ Understanding the text editors to be used ▪ Understanding the GO compiler ▪ Installation and verification of the compiler ▪ Executing the HelloWorld program with GO ▪ Basic understanding of syntax such as Tokens, Line Separator, Comments, Identifiers, Keywords, Whitespace
	Datatypes in GO	▪ Integers ▪ Standard int Type ▪ Conversion of Types ▪ Floating Point Numbers ▪ Converting float to int and back again ▪ Complex Numbers ▪ Booleans ▪ Strings Constants
Day 2	Composite Types in GO	▪ Arrays ▪ Slices ▪ Maps ▪ Structs ▪ Nested Structs
	Functions in GO	▪ Function Declaration ▪ A Simple Example ▪ Full Source Code ▪ Multiple Results From a Function ▪ Anonymous Functions ▪ Defining a Function ▪ Calling a Function ▪ Returning Multiple Values from Function

		▪ Function Arguments ▪ Call by Value, Call by Reference ▪ Function Usage ▪ Function Closures ▪ Method
Day 3	Pointers in GO	▪ What Are Pointers? ▪ How to Use Pointers? ▪ Nil Pointers in Go ▪ Go Pointers in Detail ▪ Go – Array of Pointers ▪ Go – Pointer to Pointer ▪ Go – Passing Pointers to Functions
Day 4	Variables in GO	▪ Variable Definition in Go ▪ Static Type Declaration in Go ▪ Dynamic Type Declaration / Type Inference in Go ▪ Mixed Variable Declaration in Go ▪ The lvalues and the rvalues in Go
Day 5	Operators in GO	▪ Arithmetic Operators ▪ Relational Operators ▪ Logical Operators ▪ Bitwise Operators ▪ Assignment Operators ▪ Miscellaneous Operators Operators Precedence in Go
Day 6	Decision making and Loops in GO	▪ The if Statement ▪ The if…else Statement ▪ Nested if Statement ▪ The Switch Statement ▪ The Select Statement ▪ The if...else if...else Statement ▪ for Loop ▪ Nested for Loops ▪ Loop Control Statements ▪ The continue Statement ▪ The goto Statement. ▪ The Infinite Loop
	Modules in GO	▪ Why Go Modules? ▪ The Problem ▪ A Simple Example ▪ Adding Dependencies To Your Project ▪ Handling Major and Minor Versions

Day 7	Interfaces in GO	▪ Basic Example ▪ Why is this Useful? ▪ Defining Interfaces ▪ Return Values ▪ Satisfying Interfaces
	The Init function in GO and Garbage Collector	▪ The init Function ▪ Multiple Packages ▪ Order of Initialization ▪ Multiple Init Functions in the Same File ▪ Garbage Collector
Day 8	Maps in GO	▪ The Map Data Structure ▪ Map Basic Syntax ▪ Iterating over Keys and Values ▪ Deleting Elements in a Map ▪ Mapping Strings to Interfaces ▪ Concurrency in GO ▪ What Are Goroutines? ▪ A Simple Sequential Program ▪ Making our Program Asynchronous ▪ Anonymous Goroutines Function
	Project structure in GO	▪ Small Applications - Flat Structure ▪ Benefits ▪ Examples of This Structure ▪ Medium/Large Sized Applications - Modularization ▪ Examples of This Structure ▪ Mature Projects ▪ Splitting Up
Day 9	GO Mutex	▪ The Theory ▪ A Simple Example ▪ Avoiding Deadlock ▪ Ensure You Call Unlock()! ▪ Calling Lock() Twice ▪ Semaphore vs Mutex
	GO Channels	▪ The Theory ▪ A Simple Example ▪ Unbuffered Channels ▪ Buffered Channels
	GO Waitgroup	▪ Understanding WaitGroups ▪ The Solution? - WaitGroups ▪ A Simple Example ▪ Anonymous Functions ▪ A “Real” World Example

Day 10	Introducing GO Dep	▪ Installation ▪ dep init ▪ Creating a New Project ▪ Gopkg.toml ▪ Gopkg.lock ▪ The vendor/ Directory ▪ Helpful Commands ▪ dep ensure ▪ Adding Dependencies ▪ Updating Dependencies ▪ dep status
	Microservices - Consuming a RESTful API With Go - Basics	▪ Querying The API ▪ GET Request ▪ Creating a Struct ▪ Unmarshalling our JSON ▪ Listing All Our values
	Creating a RESTful API with Golang - Basics	▪ REST Architectures ▪ JSON ▪ Marshalling ▪ Getting Started with A Basic API
Day 11	Web server with Golang	▪ Creating a Basic Web Server ▪ Running Our Server ▪ Adding a bit of Complexity ▪ Serving Static Files ▪ Checking it Works ▪ Serving Content from a Directory ▪ Checking it Works ▪ Serving Content over HTTPS ▪ Generating Keys
	GO with MySQL	▪ Connection Pooling ▪ Implementation ▪ Connecting to a MySQL database ▪ Performing Basic SQL Commands ▪ Populating Structs from Results ▪ Querying a Single Row
	Closure in GO	▪ Closures - The Theory ▪ A Simple Example ▪ Conclusion
Day 12	GO Variadic Functions	▪ Variadic Functions ▪ Production Examples
	Working with environment Variables in GO	▪ Handling Credentials ▪ Reading Environment Variables ▪ Setting Environment Variables ▪ Feature Flags

		▪ Setup Of Environment Variables
Day 13	Testing in GO - Basics	▪ Go Test Files ▪ A Simple Test File ▪ Running Our Tests
	TDD and BDD in Golang	▪ Overview of Golang Unit Testing ▪ What is Test-Driven Development (TDD) in Golang? ▪ 4 Advantages of Test-Driven Development (TDD) ▪ Test Driven Development (TDD) Process Cycle ▪ Acceptance Test Driven Development (ATDD) ▪ Behavior Driven Development (BDD) ▪ Behavior Driven Development (BDD) Approach ▪ Test Driven Development (TDD) with Golang ▪ Golang Testing for Test Driven Development using Package Testing ▪ Setting Up Environment For Test Driven Development (TDD) ▪ Implementing Test Driven Development (TDD) in Golang ▪ Summary
Day 14 and 15	Building API using OpenAPI specification – Brief Demo – Basics	▪ Tools for implementing a Golang API server with auto-generated code and documentation, grpc-gateway — protobuf, gRPC, API, WebSocket. ▪ What is OpenAPI and Swagger? ▪ OpenAPI specification ▪ Automatic generation of an OpenAPI file from code annotations ▪ Development by specification ▪ Generating Code - client or server ▪ Swagger Codegen - A tool for code generation (clients and servers) in many different languages.

		▪ go-swagger - Generates go code, and will be the subject of a following post.
Day 16 and 17	Golang ORM	▪ A Simple Example ▪ Our API ▪ SQL Database Creation and Automatic Schema Migration ▪ Updating our All-Users Endpoint ▪ Updating our New User Endpoint ▪ Our Delete User Endpoint ▪ Our Update User Endpoint ▪ Full Source Code
Day 18, 19 and 20	Gin-Gonic Framework	▪ Gin Web Framework ▪ Installation ▪ Gin v1. stable ▪ API Examples ▪ Using GET, POST, PUT, PATCH, DELETE and OPTIONS ▪ Parameters in path ▪ Querystring parameters ▪ Another example: query + post form ▪ Map as querystring or postform parameters ▪ Upload files ▪ Single file ▪ Multiple files ▪ Model binding and validation ▪ Custom Validators ▪ Only Bind Query String ▪ Bind Query String or Post Data ▪ Bind Uri ▪ Bind Header ▪ Bind HTML checkboxes ▪ Serving static files ▪ Serving data from file ▪ Serving data from reader ▪ HTML rendering ▪ Using BasicAuth() middleware ▪ Goroutines inside a middleware