Certification Program

in

GoLang Full Stack

by

GapReduce

Contact us @

GapReduce Education Services Private Limited

Website: www.gapreduce.com
Contact: +91-8602131878
Email: contact@gapreduce.com

Certification Program in GoLang Full Stack

This is the program where you will be trained with GoLang Full Stack technologies. The curriculum contains two sections - Core and Advanced namely. The entire content provides you with the comprehensive theory and practical knowledge driven by highly qualified professionals from the GloLang Full Stack background.

With the help of our carefully curated training programs, you will be professionally groomed and have your skill sets enhanced. This will be done through hands-on projects, practical assignments, case studies, communication skills, resume writing sessions, mock tests, assessments and interviews.

Features of program

- Live 2 Hours/day training sessions by GapReduce certified Trainer
- Case studies and projects
- Module wise assessments
- Practical Assignments
- Resume writing sessions
- One to One Mock interviews
- MNC specific Interview Question bank
- Job Application support
- Placement support from GapReduce Placement Team

Important Dates

Last date to Apply	November 4, 2022
Program start date	November 7, 2022
Completion of program	February 3, 2023

Program Modes

Certification program in GoLang Full Stack is offered in two modes mentioned below:

Interview Guaranteed

In GapReduce Interview Guaranteed programs, we offer guarantee for interview in respective domains covered by program. During this time, even after completion of course, learner will be given complete support to update resume, prepare profiles like linkedIn for job opportunities. Learner will be given regular updated about job opportunities and complete assistance in applying for the same.

Job Guaranteed

In GapReduce Job Guaranteed programs, we offer guarantee till the placement of learner. Learner will start receiving job updates after 70 % completion of course and this support will be given for a period of one year after completion of course. During this time, learners will be given regular updates, interview preparation question banks, mock interview and resume update sessions.

*Interview Preparation

On the basis of type of mode selected for the program, separate interview preparation modules will also be mapped to the program.

Curriculum (Duration: 122 Hours)

I. Core GoLang (72 Hours)

Module 1: Introduction to Go programming Language (4 Hours)

- The Origins of Golang
- Why to learn Golang
- Installing Go on your computer
- First Go program
- Compiling Go code
- Command-Line Arguments

Module 2: Program Structure (4 Hours)

- Names
- Declarations
- Variables
- Assignments
- Type Declarations
- Packages and Files
- Scope

Module 3: Basic Datatypes (4 Hours)

- Integers
- Floating-Point Numbers
- Complex Numbers
- Booleans
- Strings
- Constants

Module 4: Composite Types (8 Hours)

- Arrays
- Slices
- Maps
- Structs

Module 5: Functions (10 Hours)

- Function Declarations
- Recursion
- Multiple Return Values
- Errors
- Function Values
- Anonymous Functions
- Variadic Functions
- Deferred Function Calls
- Panic and Recover

Module 6: Method (4 Hours)

- Method Declarations
- Methods with a Pointer Receiver
- Composing Types by Struct Embedding
- Method Values and Expressions
- Encapsulation

Module 7: Interfaces (6 Hours)

- Interfaces as Contracts
- Interface Types
- Interface Satisfaction
- Interface Values
- Type Assertions

Module 8: Goroutines and Channels (10 Hours)

- Goroutines
- Channels
- Multiplexing with select
- Concurrency with Shared Variables
- Race Conditions
- Mutual Exclusion: sync.Mutex
- Read/Write Mutexes: sync.RWMutex

Module 9: Memory Synchronization (4 Hours)

- The Race Detector
- Concurrent Non-Blocking Cache
- Goroutines and Threads

Module 10: Packages and the Go Tool (4 Hours)

- Introduction
- Import Paths
- The Package Declaration
- Packages and Naming
- The Go Tool

Module 11: Testing (4 Hours)

- The go tests Tool
- Test Functions
- Coverage
- Benchmark Functions
- Profiling
- Reflection
- Why Reflection?
- reflect.Type and reflect.Value

Module 12: Data Structure Using Go (10 Hours)

- Array

- Slices
- Linked list
- Tree
- Graph
- Map
- Dynamic programming

II. Advanced Go (Building Distributed Applications in Gin) (50)

Module 1: Inside the Gin Framework (4 Hours)

- Introduction to Git and Git Lab
- Getting Started with Gin
- What is Gin?
- Integrated development environment (VS code setup)
- Installing Go tools
- Installing and configuring Gin
- Dependency management in Golang
- Writing a custom HTTP handler

Module 2: Distributed Microservices (10 Hours)

- Setting Up API Endpoints
- Exploring API functionality
- Defining the data model
- HTTP endpoints
- Implementing HTTP routes
- Introduction to postman and setting up postman for API Response
- Writing the OpenAPI Specification
- Installing Go Swagger
- Swagger metadata

Module 3: Managing Data Persistence with MongoDB (10 Hours)

- Running a MongoDB Server
- Installing Docker
- Running a MongoDB container
- Configuring Go MongoDB driver
- Exploring MongoDB queries
- Designing the project outline
- Caching an API using redis

Module 4: Building API Authentication (6 Hours)

- API keys
- Introducing JWT
- Sign-in handler
- Persisting client sessions and cookies
- Authenticating with Auth0
- Building an HTTPS server

Module 5: Serving HTML in Gin (8 Hours)

- Rendering HTML templates
- Creating the view templates
- Building a self-contained web application
- Getting started with React
- Exploring React components

Module 6: Scaling a Gin Application (4 Hours)

- Exploring patterns
- Scaling horizontally with Docker replicas
- Caching assets with HTTP cache headers
- Setting HTTP caching headers

Module 7: Testing Gin HTTP Routes and Deploying on AWS (8 Hours)

- Testing Gin HTTP handlers
- Performing integration tests with Docker
- Discovering security vulnerabilities
- Deploying on EC2 instance
- Launching an EC2 instance
- SSL offloading with an application load balancer
- Deploying on Amazon ECS
- Creating an ECS cluster
- Deploying on Kubernetes with Amazon EKS
- Configuring kubectl
- Migrating a Docker Compose workflow to Kubernetes