www.logicalimagination.com
800.657.1494

Go (Golang) Fundamentals

Course #: GL-100 **Duration:** 3 days

Prerequisites

Prior programming experience in at least one high-level language such as Python, Java, C#, or JavaScript, basic understanding of variables, control structures, and functions, and familiarity with command-line tools and general software development workflows. Knowledge of web APIs or SQL databases is helpful but not required.

Details

This three-day, instructor-led course provides a comprehensive introduction to the Go programming language (Golang), combining core language fundamentals with hands-on application development. Students will learn how to write efficient, maintainable Go code using modern Go tools, idioms, and best practices.

Starting with basic syntax, data types, and program structure, participants will progress into advanced concepts such as structs, interfaces, generics, and concurrency with goroutines and channels. The course concludes with a fully functional RESTful API connected to a database using the Gin framework.

Through a blend of lectures, demonstrations, and lab exercises, learners will gain the skills to design, implement, and deploy real-world Go applications confidently.

Software Needed

- **Operating System**
  - Windows 10 or later
  - macOS (current or recent version)
  - Linux (any modern distribution)
- **Go (Golang)**
  - Latest stable release of Go (installed prior to class)
  - Available from the official Go website
- **Code Editor or IDE** (one of the following recommended)
  - Visual Studio Code (with Go extension)
  - GoLand
  - Any text editor with Go language support
- **Web Browser**
  - Modern browser such as Chrome, Edge, or Firefox
  - Used for documentation, testing APIs, and learning resources
- **Git**
  - For source control and working with sample projects
- **REST Client**
  - Postman, curl, or equivalent tool for testing API endpoints

Outline

Go (Golang) Fundamentals

- **Introduction to Go**
  - What is Go
  - Installing Go and Setting up an Editor
  - Writing a First Go Program
  - Using Go Modules
- **Go Essentials**
  - Variables, Values, and Operators
  - Strings and Type Conversions
  - Control Structures
  - Functions and Error Handling
  - Working with Files
- **Organizing Code**
  - The main Function and Packages
  - Importing Packages and Exporting Code
  - Using Third-Party Packages
  - Go Tooling: fmt, vet, doc
- **Understanding Pointers**
  - Writing Code Without Pointers
  - Creating a Pointer
  - Passing Pointers to a Function
  - Using Pointers for Data Manipulation
- **Structs and Custom Types**
  - Defining a Struct Type
  - Passing Structs as Arguments
  - Structs and Pointers
  - Methods and Constructor Functions
  - Structs, Packages, and Exports
- **Interfaces and Generic Code**
  - Interface Use-Case
  - Creating and Using Interfaces
  - Embedded Interfaces
  - Type Switches and Dynamic Types
  - Introducing Generics
- **Working with Collections**
  - Introducing Arrays
  - Using Slices of Arrays
  - Introducing Maps
  - Mutating Maps
  - Maps vs. Structs
  - Looping with Arrays, Slices, and Maps
- **Deep Dive into Functions**
  - Functions as Values
  - Anonymous Functions
  - Understanding Closures
  - Recursion
  - Variadic Functions
  - Splitting Slices into Parameter Values
- **Concurrency**
  - Introducing Goroutines
  - Running Functions as Goroutines
  - Using Channels for Communication
  - Setting up an Error Channel
  - Managing Channels with Select
  - Defer, Panic, and Recover
- **Building a REST API**
  - Gin Framework Setup
  - Setting up a Model
  - JSON, Middleware, and Context

    - Registering a POST Route
    - Testing Requests

- **Working with Databases**
    - SQL Integration and Queries
    - CRUD Operations and Transactions
    - Preparing Statements
    - Refactoring for Clean Architecture