

Go Programming in 24 Hours

Duration: 3 Days

Prerequisites: Knowledge of any programming language

Day 1: Go Foundations and Core Syntax

Theme: Build a strong foundational base for Go language syntax, structure, and workflow.

Session 1: Introduction to Go & Environment Setup

- Why Go? Benefits and Real-World Use Cases (Docker, Kubernetes, Web Services)
- Installing Go and Understanding GOPATH, GOROOT, and Modules
- Setting up VS Code or GoLand
- Writing, Running, Building, and Formatting Go Code

Lab 1: Setup and Hello Go

- Create a "Hello, Go!" project with proper module structure
- Practice formatting, compiling, and running using CLI

Session 2: Core Language Constructs

- Variables, Constants, Type Inference
- Primitive Data Types
- Operators and Expressions
- Control Flow: if, switch, for, range

Lab 2: Flow Control Exercises

- Build a simple calculator app using user input and control statements
- Write a program to print patterns and prime numbers

Session 3: Functions and Error Handling

- Declaring and Calling Functions
- Multiple Return Values and Named Returns
- Variadic Functions
- Error Handling using error type
- Introduction to defer, panic, and recover

Lab 3: Function Practice

- Write utility functions for string manipulation and math operations
 - Simulate basic error scenarios and graceful handling
-

Day 2: Data Structures and Program Structure

Theme: Dive into Go's memory model, data types, and structured programming using collections and custom types.

Session 1: Arrays, Slices, and Maps

- Arrays: Basics, Memory Layout
- Slices: Creation, Capacity, Append, Copy
- Maps: Create, Read, Update, Delete
- make, new, and built-in functions

Lab 4: Collections Challenge

- Implement a frequency counter using maps and slices
- Build a grade management system (CRUD-like logic)

Session 2: Structs and Methods

- Structs: Definition, Initialization, Tags
- Embedding and Composition
- Methods and Receiver Types (Value vs Pointer)
- Introduction to Custom Packages

Lab 5: Struct-Based Mini App

- Design a Book struct with methods to manage a library
- Add search and update features

Session 3: Interfaces and Modular Design

- What is an Interface? Why use them?
- Interface Implementation, Type Assertion, Type Switch
- Empty Interface and Use Cases
- Organizing Code into Packages and Modules

Lab 6: Interface in Action

- Create a shape interface with multiple implementations (Circle, Rectangle)
- Build a simple app using interfaces for pluggability

Day 3: Concurrency, File I/O, and Capstone

Theme: Introduce Go's signature concurrency model, basic I/O, and bring everything together in a hands-on project.

Session 1: Concurrency with Goroutines and Channels

- What is Concurrency vs Parallelism?
- Goroutines: Syntax and Behavior
- Channels: Unbuffered and Buffered
- Channel Operations and select Statement
- Synchronization with WaitGroups

Lab 7: Concurrent App

- Create a concurrent web fetcher (fetch multiple URLs and print size)
- Build a concurrent prime number generator

Session 2: File Handling and JSON

- Reading from and Writing to Files
- Working with JSON: Encoding/Decoding Structs
- io, os, and encoding/json packages

Lab 8: File-Based Logger

- Create a log system that writes logs to a file
- Parse JSON configuration into a Go struct