

Go Programming Fundamentals (Go Language)

Prerequisites: Knowledge of any programming language

Day 1: Introduction to Go and Core Concepts

Objective: Familiarize participants with Go basics, control structures, and data types.

- **Morning Session (Theory)**
 1. **Introduction to Go Programming Language**
 - History, features, and advantages of Go
 - Why use Go? Applications and use cases
 2. **Setting Up the Development Environment**
 - Installing Go on local machines
 - Setting up the Go workspace and Go modules
 - Introduction to Go tools (go run, go build, go fmt)
 3. **Basic Go Syntax and Structure**
 - Packages, imports, and main function
 - Variables and constants
 - Data types (integers, floats, strings, booleans)
- **Afternoon Session (Lab)**
 1. **Lab 1: Go Installation & Hello World**
 - Install Go and configure the workspace
 - Write and run a simple "Hello, World" program
 2. **Lab 2: Variables and Basic Data Types**
 - Create variables of different types (integers, strings, etc.)
 - Experiment with basic I/O: reading input and printing output
- **Evening Session (Theory)**
 1. **Control Structures in Go**
 - if, else, else if conditions
 - switch statements

- Loops (for, range, break/continue)
 - **Lab 3: Control Structures Practice**
 - Write a simple number guessing game using if-else and for loops
 - Implement a basic menu-driven application using switch statements
-

Day 2: Functions, Arrays, and Slices

Objective: Build upon the basics with functions, error handling, arrays, and slices.

- **Morning Session (Theory)**
 1. **Functions in Go**
 - Defining functions and parameters
 - Return values and multiple returns
 - Understanding pass-by-value
 2. **Error Handling**
 - Using Go's built-in error handling (multiple return values)
 - Introduction to the errors package
- **Lab 4: Function Practice**
 - Create a function to calculate the sum of two numbers
 - Create a function that returns both a result and an error
 - Use error handling in a program that divides two numbers
- **Afternoon Session (Theory)**
 1. **Arrays and Slices**
 - Defining arrays and slices
 - Differences between arrays and slices
 - Common slice operations: append, copy, slicing
- **Lab 5: Arrays and Slices Manipulation**
 - Create arrays and iterate over them with loops
 - Use slices to manipulate dynamic data

- Build a program to take a list of numbers, append values, and calculate averages
- **Evening Session (Lab & Wrap-Up)**
 1. **Lab 6: Mini Project - Building a Basic To-Do List**
 - Create a to-do list application using functions, slices, and control structures
 - Add, view, and delete tasks from the list
 - Use error handling for invalid inputs