



Golang for Beginners



Go หรือ Golang เป็นภาษาโปรแกรมมิ่งที่มีการพัฒนาโดย Google และถูกออกแบบมาเพื่อให้มีประสิทธิภาพสูง มีความปลอดภัย และเหมาะสำหรับการพัฒนาโปรแกรมแบบ Concurrent หรือการทำงานพร้อมกันหลายเธรด (Concurrency) ซึ่งเป็นส่วนหนึ่งของความเป็นภาษาซึ่งมีความเป็นเนื้อหาเด่นของ Golang

Go ถูกออกแบบขึ้นมาในปี 2009 โดย Rob Pike, Ken Thompson, และ Robert Griesemer ซึ่งเป็นผู้พัฒนาที่มีชื่อเสียงในโลกโปรแกรมมิ่ง ความเร็วในการคอมไพล์ Go สูง และช่วยให้โปรแกรมที่เขียนด้วย Go มีประสิทธิภาพสูงในการทำงาน โดยเฉพาะการทำงานกับเว็บ

ภาษา Go เป็นภาษาแบบสามารถคอมไพล์ได้และเป็นภาษาแบบ Statically typed ซึ่งหมายความว่าต้องประกาศตัวแปรก่อนใช้งาน และมีเครื่องมือช่วยการจัดการข้อผิดพลาด (Error Handling) ที่ดี โดยเซตของฟังก์ชันและการใช้งานของโค้ดจะเขียนในรูปแบบของ Module ที่ช่วยให้การจัดการโค้ดและการแบ่งแยกโค้ดมีความสะดวกสบายและเป็นระเบียบมาก

วัตถุประสงค์:

- สามารถพัฒนาโปรแกรมแบบ Concurrent หรือการทำงานพร้อมกันหลายเธรดได้

กลุ่มเป้าหมาย:

- ผู้ที่ต้องการพัฒนาเว็บไซต์ที่รองรับ Request เป็นจำนวนมากได้

ความรู้พื้นฐาน:

- พื้นฐานการเขียนโปรแกรม

ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)

ราคาคอร์สอบรม

- 8,500 บาท / คน (ราคานี้ยังไม่ได้รวมภาษีมูลค่าเพิ่ม)

วิทยากรผู้สอน:

- อาจารย์สนิทวงศ์ กมลภากรณ์



เนื้อหาการอบรม:

Module 1: Introduction to Go

- Structure
- Objective
- A look at software essence
- Evolution of Golang and its success
- Characteristics of Go
- Object-oriented nature
- The analogy with other languages
- Pain points of Go language
- Applications of Golang

Module 2: Environment Setup

- Structure
- Objective
- System requirements
- How to install
- Uninstalling Go

Module 3: Beginning With Go

- Structure
- Objective
- Introduction
- Keywords
- Identifiers

Module 4: Variables, Data Types, and Constants

- Structure
- Objective
- Variables
- Data types
- Constants
- Format specifiers



Module 5: Operator

- Structure
- Objective
- Introduction
- Types of operators
- Operator precedence

Module 6: Control Structures

- Structure
- Objective
- Introduction
- Branching statements
- Looping statements
- Jumping statement

Module 7: Functions

- Calculating Area of Triangle using Arithmetic Operators
- Calculate the Arithmetic Function
- Example of Operators
- Example of Operators Continues

Module 8: Returning Multiple Values from Functions

- Structure
- Objectives
- Introduction
- Declaring functions
- Function arguments and return types
- Calling a function
- Call by value
- Call by reference
- Named return values
- Blank Identifier
 - Using a function as a type
- Anonymous functions
 - Return Anonymous functions



- Closures
- Recursive functions
- The defer keyword
- Variadic function

Module 9: Arrays and Slices

- Structure
- Objective
- Introduction
 - Declaring an array
 - Initializing an array
 - Need for an array
 - Multidimensional arrays
 - Functions of an array
- Slices
 - Declaring slices
 - Initializing slices
 - Slices from an array
 - Slice from slice
 - Slice from structs
 - Slice internals
 - Slice using the make function
 - Two-dimensional slices
 - Slice functions

Module 10: Strings

- Structure
- Objectives
- Introduction
 - Functions with strings
- The rune data type
 - Iterate using range keyword
 - Convert a slice of runes to string
 - Convert a slice of bytes to string



- Immutable nature of the string
- Strings package
- Functions with strings package

Module 11: Pointers

- Structure
- Objectives
- Introduction
- Need for pointers
- Declare pointers
- Initializing pointers
- Dereferencing pointers
- Manipulating pointers
- Creating pointers with new
- Nested pointers
- Array of pointer
- Passing pointers to a function

Module 12: Structures

- Structure
- Objectives
- Introduction
- Declaring struct
- Initializing struct
- Accessing struct's field
- Pointer to struct
- A struct with a new function
- Anonymous struct fields
- Nested struct
- Passing structs to functions/copying structs
- Comparing structs



Module 13: Composition

- Structure
- Objectives
- Introduction
- Embedding pointers
- Embedding slice of structs

Module 14: Interfaces and Polymorphism

- Structure
- Objectives
- Introduction
- Declaring Interface
- Implementing interface
- Empty interface
- Implementing multiple interfaces
- Type assertion
- Type switch
- Embedding interfaces
- Interfaces as polymorphism

Module 15: Maps

- Structure
- Objectives
- Introduction
 - Use of maps
- Declaring a map
- Initializing a map
- Map operations
- Map as a reference type



Module 16: Concurrency with Go

- Structure
- Objectives
- Introduction
- How parallelism is different from concurrency
 - Thread
- Goroutines
 - Multiple Goroutines
 - Goroutines at multiple cores
- Waiting for Goroutine to finish

Module 17: Mutex and Channels

- Structure
- Objectives
- Declaring channels
- Channel operations
- Deadlock
- Closing a channel
- The direction of a channel
- The select keyword
- Data race in Go
 - Race condition
 - Check for data race
- Mutex
 - RWMutex

Module 18: Error Handling

- Structure
- Objectives
- Introduction
- Error package
- Defer statement
- Panic
- Recovery



Module 19: Reflection

- Structure
- Objectives
- Introduction
- The reflect package
 - Type
 - Value
 - Kind
 - NumField()
 - Field
 - String(), Int() & Float()
- Laws of reflection

Module 20: Building Web Application

- Structure
- Objectives
- Introduction to web development
- HTML
- JavaScript
 - Internal JavaScript
 - External JavaScript
- CSS
- GopherJS
- Database
- Building an applicationn from the browser