



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-1201L

Introduction to Information and Communication Technology Fall 2019

LAB 10 – Introduction to GO Language

OBJECTIVE(S)

- Learn about Introduction to Go Programming language
- Learn about the purpose for developing Golang
- Learn about the features of Golang
- Learn about Programming in Golang

Introduction to GO Programming Language

Go Programming language is a procedural and statically typed programming language having the syntax similar to C programming language. It is often referred as Go Programming Language. It provides a rich standard library and garbage collection capability. It was developed in 2007 by Robert Griesemer, Rob Pike, and Ken Thompson at Google but launched in 2009 as an open-source programming language and mainly used in Google's production systems. Golang is one of the most trending programming languages among developers.

Why Golang?

The authors mention that the primary motive for designing a new language was to solve software engineering issues at Google. They also mentioned that Go was actually developed as an alternative to C++.

The main purpose of designing Golang was to eliminate the problems of the existing languages. The problems that were faced with Python, Java, C/C++ languages were:

- **Python:** "Easy to use, but slow". Python is easy but slow as it is an interpreted language that makes the code slow.
- **Java:** "Complex type system". Java has a complex type system due to its additional features which makes it more complex and slow.
- **C/C++:** "Complex type system as well as slow compilation time". C/C++ was designed when the computer does not have that much space so the developer decided to optimize the compiler by using the minimal amount of space which results in sluggish compiler time.

Also, all these languages were designed when multi-threading applications were rare, so these are not much effective to highly scalable, concurrent and parallel applications.



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-1201L

Introduction to Information and Communication Technology Fall 2019

Features of Golang

- **Binaries:** Go generates binaries for your applications with all the dependencies built-in. This removes the need for you to install runtimes that are necessary for running your application. This eases the task of deploying applications and providing necessary updates across thousands of installations. With its support for multiple OS and processor architectures, this is a big win for the language.
- **Language Design:** The designers of the language made a conscious decision to keep the language simple and easy to understand.
- **Powerful standard library:** Go comes with a powerful standard library, distributed as packages.
- **Static Typing:** Go is a statically typed language. This means that you need to declare types for all your variables and your function arguments (and return variables) at compile time. Although this may sound inconvenient, this is a great advantage since a lot of errors will be found at compile time itself. This factor plays a very big role when your team size increases, since declared types make functions and libraries more readable and easier to understand. This can avoid the problems that one faces in dynamically typed languages, where you discover the issues only when the code is executed.
- **Concurrency Support:** Go has first class support for concurrency. Concurrency is one of the major selling points of Go. The Go runtime allows you to run hundreds of thousands of concurrent go procedures on a machine.
- **Testing Support:** Go Language brings Unit Testing right into the language itself. It provides a simple mechanism to write your unit tests in parallel with your code.
- **Garbage collection:** Automatic garbage collection of Go is very fast.

Golang Programming

Since Golang is a lot similar to other widely used languages syntactically, it is easier to code and learn in Golang. Programs can be written in Golang in any of the widely used text editors like Notepad++, etc. or on any of the text-editors. After writing the program, save the file with the extension .go or .GO.



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-1201L

Introduction to Information and Communication Technology Fall 2019

(i) Example Program (Print Hello World):

```
package main
import "fmt"
func main() {
    fmt.Println("Hello World!")
}
```

Explanation of the syntax:

Line 1 (Package Declaration): Every Go program starts with a package declaration. Packages are used to organize related go source code files into a single unit and make them reusable. The package “main” is a special go package that is used with programs that are meant to be executable.

Line 2 (Import Statement): The import keyword is a preprocessor command which tells the compiler to include the files lying in the package in our program. The “fmt” package contains code for dealing with I/O.

Line 3 (Function Declaration): The main() function is the entry point of an executable program in Go. It is the first thing that is invoked when you run an executable program.

Line 4 (Call to the Println() function): This line contains a call to the Println() function of the fmt package. It is a standard library function to print output on screen.

(ii) Example Program (Double the given number):

```
package main
import "fmt"
func main() {
    fmt.Print("Enter a number: ")
    var input float64
    fmt.Scanf("%f", &input)
    output := input * 2
    fmt.Println(output)
}
```



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-1201L

Introduction to Information and Communication Technology Fall 2019

(iii) Example Program (Find even or odd numbers in a loop):

```
package main
import "fmt"
func main() {
    for i := 1; i <= 10; i++ {
        if i % 2 == 0 {
            fmt.Println(i, "even")
        } else {
            fmt.Println(i, "odd")
        }
    }
}
```

(iv) Example Program (Displaying remarks using Switch statement):

```
package main
import "fmt"
func main() {
    fmt.Print("Enter a number: ")
    var input int64
    fmt.Scanf("%d", &input)
    switch input {
    case 0: fmt.Println("Very Bad")
    case 1: fmt.Println("Unsatisfactory")
    case 2: fmt.Println("Satisfactory")
    case 3: fmt.Println("Good")
    case 4: fmt.Println("Very Good")
    case 5: fmt.Println("Excellent")
    default: fmt.Println("Unknown Number")
    }
}
```



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-1201L

Introduction to Information and Communication Technology
Fall 2019

(v) Example Program (Performing basic mathematical operations):

```
package main
import "fmt"
func main() {
    var x, y = 20, 15
    x += y
    fmt.Println("Addition=", x)
    x -= y
    fmt.Println("Subtraction=", x)
    x *= y
    fmt.Println("Multiplication=", x)
    x /= y
    fmt.Println("Division=", x)
    x %= y
    fmt.Println("Modulus=", x)
}
```



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-1201L

Introduction to Information and Communication Technology Fall 2019

LAB ASSIGNMENT

1. Write a program that reads the distance between two cities (in kilometers) and converts the distance into meters and feet.
2. Write a program that checks whether the number entered by user is negative, positive or zero.
3. Write a program to make a Simple Calculator to add, subtract, multiply or divide numbers using switch case.
4. Write a program that accepts a three digit numerical input from the user and checks whether it is an Armstrong number or not.
5. Write a program to calculate the sum of first n natural numbers. The value of n must be provided by the user.

SUBMISSION GUIDELINES

- Take a screenshot of each task.
- Place all the screenshots in a single word file labeled with Roll No and Lab No. e.g. 'cs191xxx_Lab01'.
- Convert the file into PDF.
- Place all the related files along with the PDF file in a folder labeled with Roll No and Lab No. e.g. 'cs192xxx_Lab01'.
- Submit the folder at [LMS](#)
- -100% policies for plagiarism.