

Starting Go

Divyashree R

<https://www.linkedin.com/in/divyashree-r-aaa645166/>

Saturday, June 27, 2020 8:07 PM

Part 1

- Example 1 --> Covers basic Go code, how to compile and run
- Example 2--> covers Variable assignments

Example 1:

Check the below code "welcome.go"

1. Imports package "fmt"
2. In main, prints the line "Welcome to GO Tutorials"

```
// Basic code to demonstrate GO code

package main

import (
    "fmt"
)

func main() {
    fmt.Println("Welcome to GO Tutorials")
}
```

Follow the below steps:

1. To compile and run the code:

```
PS E:\GoCode\Go_Tutorials> go run welcome.go
Welcome to GO Tutorials
```

2. To build the code:

```
PS E:\GoCode\Go_Tutorials> go build welcome.go
PS E:\GoCode\Go_Tutorials> .\welcome.exe
Welcome to GO Tutorials
```

This generates the welcome.exe and to run ./welcome.exe

3. To clean the object files:

`go clean`

To get information on all go commands --> `go help`

```
go <command> [arguments]
```

The commands are:

bug	start a bug report
build	compile packages and dependencies
clean	remove object files and cached files
doc	show documentation for package or symbol
env	print Go environment information
fix	update packages to use new APIs
fmt	gofmt (reformat) package sources
generate	generate Go files by processing source
get	download and install packages and dependencies
install	compile and install packages and dependencies
list	list packages or modules
mod	module maintenance
run	compile and run Go program
test	test packages
tool	run specified go tool
version	print Go version
vet	report likely mistakes in packages

Example 2: Variable assignment and arithmetic operations

Code to understand variable declaration and assignment, and arithmetic operations usage

arithmetic_operations.go

```
// Sample code to explain all the arithmetic operations
package main

import (
    "fmt"
)

func main() {
    var x int    // method 1: declaring & assigning a variable
    var y int    // y is a variable of type int

    x = 1    // assigning a variable
    y = 2

    // it can also be assigned as x, y = 1, 2

    a:=5.0    // method 2: := takes type of the value assigned
    b:=2.5

    sum := x+y    // arithmetic + operator
    sub := y-x    // arithmetic - operator
    mul := x*y    // arithmetic * operator
    quotient := a / b    // arithmetic / oprator

    // All the variables used in one expression must be of same type

    fmt.Printf("Sum: %v, type of %T\n", sum, sum)    // %v--> print the go object & %T--> prints the type of the variable
    fmt.Printf("Difference: %v, type of %T\n", sub, sub)
    fmt.Printf("Product: %v, type of %T\n", mul, mul)
    fmt.Printf("Quotient: %v, type of %T\n", quotient, quotient)
}
```