Go Language - Hands on Exercises

Mumbai Tech Meetup - January 31, 2016

Development Environment Setup

1. Either run them in Go Playground
2. Or if you have a local setup, ensure that Go has been installed, and you can invoke the **go build** tool from anywhere. For e.g. try out **go version**
3. Create a folder **c:\techmummeetup-go**
4. For each hands-on exercise, create a folder. For e.g. ex1 and so on. Then in each file, create a file named **main.go** with the code.
5. For each program, go to the terminal / command prompt and then go to the folder and give **go run main.go**

You are all set !

Exercise 1: Hello World

Statement : Write a basic Hello World

Solution : http://play.golang.org/p/achXqgsH1v

Exercise 2 : Variables & Constants

Statement : Create multiple variables that model a Car. Create the following variables:

- Car Model : string
- Car Brand : string
- Car Year : string
- Car Price : float64
- Create a constant manufacturer and set it to some value
- Print out all the variables and constants in main() function

Solution : http://play.golang.org/p/dQuYzSqR0P

Exercise 3 : Create Functions

Statement : Create a couple of functions as given below:

- PrintDetails : This function will print the details of the car.
- prettyPrintPrice: This function will provide a nicely formatted price string with the currency symbol.

Solution :

- http://play.golang.org/p/Z85-wKYlTI

Exercise 4 : for , if , switch statements

Statement :

1. Write a for loop that calls the printDetails function 3 times.
   http://play.golang.org/p/wtDFYDWXQ2
2. Write a function that uses an if statement to return true if price of car is greater than 300000.
   http://play.golang.org/p/99Qotidehx
3. Write a function that uses a switch statement to check if the brand name is local or imported.
   http://play.golang.org/p/2uf_MEB4-p

Exercise 5 : Using Maps

Statement :

1. Create a map named TaxRates of type [string]float32, where the key is the state code e.g. MH, UP, etc and the value is the tax rate
2. Initialize it with with sample values
3. Write a function that takes in a car price and state code and returns back the computed tax.

   Solution: http://play.golang.org/p/pMjCGN3Tnf

Exercise 6 : Using Structs and Arrays

Statement :

1. Refactor the entire code using a struct for the Car.
2. Move the functions for printDetails, calculateTax as receiver methods for the struct
3. Create an Array of type Cars, size of Array = 3
4. Initialize it with Car instances
5. Execute the receiver methods on the Car instances

Solution: http://play.golang.org/p/TRt2wOuBkT

Exercise 7 : Composition

Statement :

1. Create new structs that embed the Vehicle Type.
2. The new structs are:
   a. BaseModel : This will embed the Vehicle Type as is
   b. MidrangeModel : This will embed Vehicle Type and an extra attribute : stereosystem
   c. PremiumModel : This will embed Vehicle Type and 2 extra attributes : acmodel, acsize
3. Create 3 methods on Vehicle Type : Start, Stop and PrintDetails
4. Override PrintDetails in both MidrangeModel and PremiumModel
5. In main program, instantiate multiple types and invoke all the methods.

Solution: http://play.golang.org/p/toycLtAVxz

Exercise 8 : Using Inheritance

Statement :

1. Create an Interface with the Start, Stop, PrintDetails methods.
2. Ensure Vehicle implements them.
3. Create a method that accepts a variable of the Interface type. Pass BaseVehicle Instance, Midrange Instance and so on to the method.

Solution: http://play.golang.org/p/i1Hpf2YA5-