



Golang - Assignment 1

Exersice 1

What command did you use to compile and run your Go program?

```
go run main.go
```

What are the basic components of a Go program (e.g., package, import, function)?

main package, main function, import libraries

How does Go handle package imports?

By folders structure, one folder -> one package

Exersice 2

What is the difference between using var and := to declare variables?

“var” declares a variable with an explicit type and can be initialized later. “:=” is short-hand of declaration and initialization

How do you print the type of a variable in Go?

Using special method of “fmt” package `fmt.Printf()` with “%T” template, where T is short-hand type. (%d - int, %.2f - float, %t - bool, %s - string)

Can you change the type of a variable after it has been declared? Why or why not?

No, because Go is compiled statically-typed language

Exersice 3

How does the if statement in Go differ from other languages like Python or Java?

Parentheses not required and you can include in condition short-hand statement

What are the different ways to write a for loop in Go?

omit condition (while true), without semicolons (while condition), and default with semicolons

How does a switch statement in Go differ from switch in languages like C or Java?

Cases break automatically, you can also make comparisons in cases, and make assignments statements

Exersice 4

How do you define a function with multiple return values in Go?

split values by commas in return statement and declare type

What is the significance of named return values in Go?

for more readability and explicit typing

How does a switch statement in Go differ from switch in languages like C or Java?

print underscore instead variable name

Exersice 5

How do you define a struct in Go?

```
16
17 < type Person struct {
18     Name string
19     Age  int
20 }
21
22 < func (p Person) Greet() {
23     fmt.Printf("Hello %s", p.Name)
24 }
```

How do methods differ from regular functions in Go?

In methods you define the specific type it belongs to

Can a method in Go be associated with types other than structs?

Yes, example for int, float32, bool

Exersice 6

What is embedding in Go, and how does it relate to composition?

Embedding allows a struct to include another struct type like extend

How does Go handle method calls on embedded types?

You can also using methods of embedded struct in parent

Can an embedded type override a method from the outer struct?

Methods of outer structs can override inner structs, but not vice versa

Exersice 7

How do you define and implement an interface in Go?

By specifying a set of methods in interface, and implement their in methods of any struct

What is the role of interfaces in achieving polymorphism in Go?

You can implements for interface for any types and structures

How can you check if a type implements a certain interface

IDE will showing error

Exersice 8

How do you work with JSON in Go?

Using package json with methods marshal, unmarshal

What role do struct tags play in JSON encoding/decoding?

Tags such “json” define fields for json serialization

How do you handle errors that may occur during JSON encoding/decoding?

Return multiple vars in func: <value>, error