17.11.2022 Go Lesson Instruction

87. Map - Introduction  
m := map[string]int{  
"Javokhir": 20,  
"Ravi": 35,  
}  
If I want to see in output other one name, that variable of value will be 0 automatically. Maps are a convenient and powerful built-in data structure that associate values of one type (the key) with values of another type (the element or value). Slices cannot be used as map keys, because equality is not defined on them  
  
88. Map - add value to map  
If I have m map and I want to add new value to map, I use this `m["John"] = 34`  
  
89. Map - deleting  
I can use just delete(m, "Javokhir")  
  
90. Struct  
type <type\_name> struct {  
first: string,  
last: string,  
}  
person := {  
first: "First name",  
last: "Last name",  
}  
  
91. Embedding Struct  
type person struct {  
first string  
last string  
}  
type AllStruct struct {  
person  
}  
These codes shown that we can embed struct, and than I will have a question. How can I use?  
ab := AllStruct {  
person: person{  
first: "Javokhir", fmt.Println(ab.person.first)  
last: "Nematov", `first` and `last` have to be unique  
}  
}  
  
92 .Anonymously Struct   
ab := struct {  
first string  
last string  
}{  
first: "Javokhir",  
last: "Nematov",  
}  
  
93. Housekeeping  
Go programming was build for effecient compilation, effecient execution and ease.  
  
var first string var, type  
= We create a VALUES of certain TYPE  
type person struct {  
last string  
}  
  
94. Syntax  
func\_name(variable type) for getting data from function.  
Everything in GO is pass by value.  
`func woo(b string) string {  
return fmt.Sprint("Hello", b)  
}` for sending data to main menu from outfit function.  
  
I here we have to pay attention to `Sprint` and `string type`  
If we get 2 data from as like this  
`func example (ab string, bc bool (string, bool)`  
  
what is iota.  
Iota in Go is used to represent constant increasing sequences. When repeated in a constant, its value gets incremented after each specification.  
const (  
first = iota  
second = iota  
third = iota  
)  
fmt.Println(first, second, third)  
  
95. Variatic Parameter  
  
func main() {  
foo(1, 2, 3, 4, 5, 6, 6, 7, 8)  
}  
func foo(x ...int) {  
fmt.Println(x)  
fmt.Printf("%T\n", x)  
}