Go is a complier language

It is case sensitive

**Variables and constants:**

* Should declare a variable: var
* Eg: **var** num **int** =2…int is the type of the variable
* In go if variable Is declared than it must be used, else there will be an error
* /\*func main() {
* //using variables
* var num1 int  = 2
* var num2 int = 3 ///uint only accepts positive value
* num3 :=8 // easy way to assign variable
* //var num1,num2 int ....varible with the same type can be assigned in the single line...
* num1,num2= 4,6 /// example of setting the variable in the same line
* const num4= 8 // using constant
* //num4=7 ...this cannot be done since num4 is declared as constant.
* fmt.Println("Hello aarav")
* fmt.Println((num1+ num2))
* fmt.Println(num3)
* }\*/

**Functions:**

The variable and output to be used should be defined inside and outside respectively of the function.

The function are called and executed inside the “main ” function.

**Defer function:**

It is used to execute the specified function at “last”.

For an example: if main() contains function b() even at the first line, “defer b()” will be executed at last.

**Exported names:**

Capital letters should be used to define functions if it is to be used outside the package (working file.)

Eg: func Demo(){}…can be exported in other packages

Why? Because Golang is case sensitive.

**Variable scope:**

Package level: declared outside any function…

function level variable: declared within the function, the package level variable can be assigned different value inside a function.

**Math package:**

Any package should be imported

For math it is “math”

**If else and switch**:

There should be open and close curly bracket after each statement.

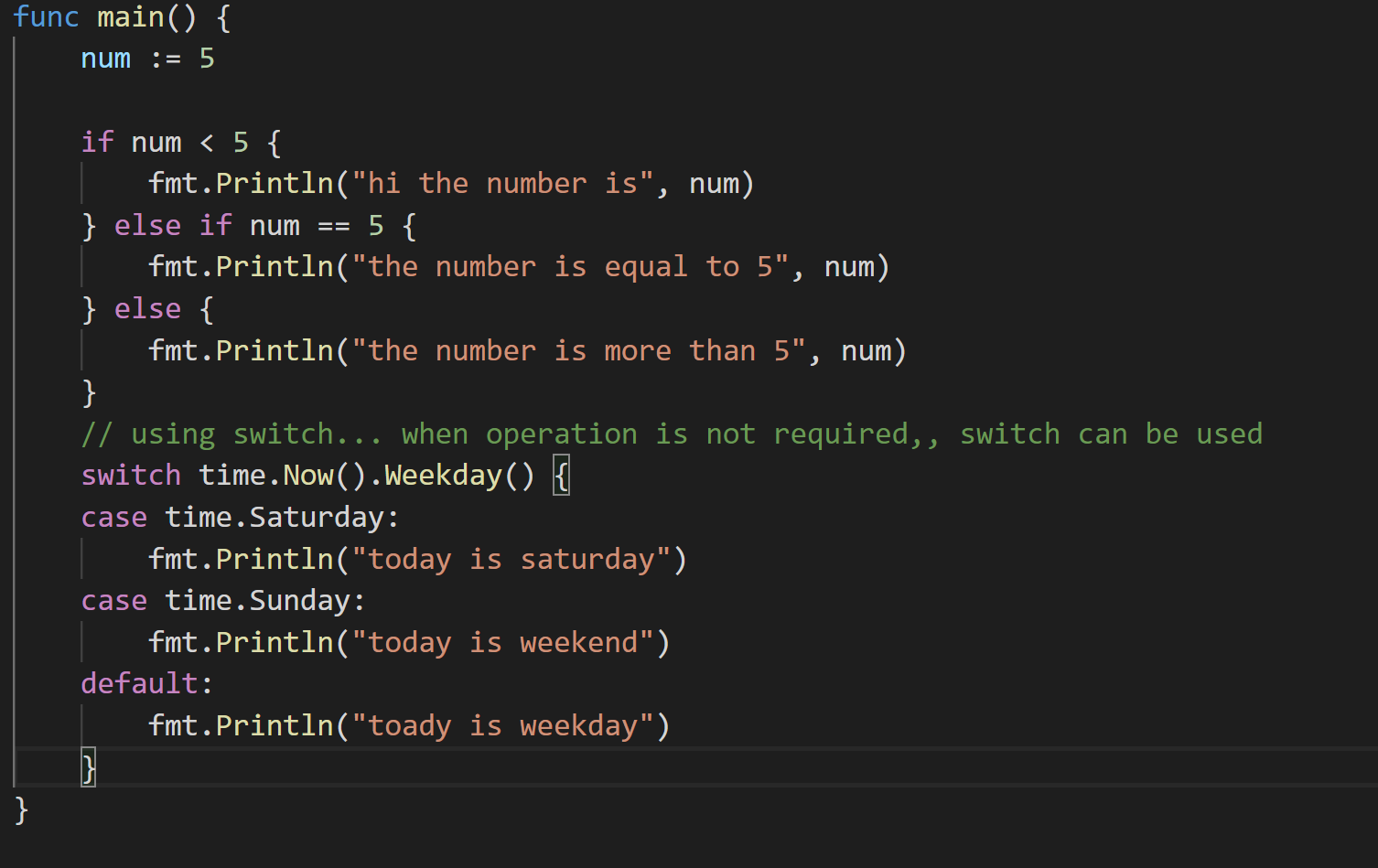
Next statement should begin after the ending bracket of preceding statement.

**Switch:**

It is easy to use when no operation is required.

The switch condition is compared with the cases.

Default is the default value after all the cases are executed.



**Structure :**

It gives the platform to assign multiple value to a single variable.