NCTU
Golang 2021

Lab 2

2021 / 9 / 30

Control Flow

1. Sequence
2. Selection
3. Iteration

Selection Structure

Comparison Operators

== equal
!= not equal
< less
<= less or equal
> greater
>= greater or equal

https://golang.org/ref/spec#Comparison_operators

Logical Operators

&& conditional AND p && q → "if p then q else false“
|| conditional OR p || q → "if p then true else q“
! NOT !p → "not p"

https://golang.org/ref/spec#Logical_operators

Example

X > 50 || X < 30

X = 20 ?

True

X = 40 ?

False

X = 70 ?

True

https://golang.org/ref/spec#Logical_operators

Example

X > 30 && X < 50

X = 20 ?

False

X = 40 ?

True

X = 70 ?

False

https://golang.org/ref/spec#Logical_operators

if-else-else if

```
if condition {
    …
}
```

if-else-else if

```
if condition {
    …
} else {
    …
}
```

if-else-else if

```
if condition {
    …
} else if condition {
    …
}
```

if-else-else if

```
if condition {
    …
} else if condition {
    …
} else {
    …
}
```

if-else-else if Example

```
if integer == 3 {
        fmt.Println(“The integer is equal to 3”)
} else if integer < 3 {
        fmt.Println(“The integer is less than 3”)
} else {
        fmt.Println(“The integer is greater than 3”)
}
```

Switch case

```
Switch sExpr {
case expr1:
    …
case expr2:
    …
default:
    …
}
```

Switch case

```
i := 10
switch i {
case 1:
	fmt.Println("i is equal to 1")
case 2, 3, 4:
	fmt.Println("i is equal to 2, 3 or 4")
	fallthrough
case 10:
	fmt.Println("i is equal to 10")
default:
	fmt.Println("All I know is that i is an integer")
}
```

Iteration Structure

for loop

```
for expression1; expression2; expression3 {
    …
}
```

for loop - example 1

```
func main {
        sum := 0
        for index:=0; index < 10; index++ {
                sum += index
        }
        fmt.Println(“sum is equal to “, sum)
}
```

for loop

```
for ; expression2;{
    …
}
```

for loop - example 2

```
func main {
        sum := 1
        for ; sum < 1000; {
                sum+= sum
        }
}
```

break, continue

```
for index := 10; index > 0; index-- {
        if index == 5 {
                break
        }
        fmt.Println(index)
}
```

Output: 10 9 8 7 6

```
for index := 10; index > 0; index-- {
        if index == 5 {
                continue
        }
        fmt.Println(index)
}
```

Output: 10 9 8 7 6 4 3 2 1

Lab 2 Sum and print it all

1. Pull or clone the project on github
2. Complete the program
3. Take a screenshot of output of your program and name it ID_Name_Lab2.jpg
4. Rename your .go file to lab2.go
5. Upload your screenshot and source code to Google Classroom
   (Don't compress them)

Gist: https://github.com/benchen216/NCTU-GoProgramming-2021

Lab 2 Sum and print it all

The input is n, please print all the numbers which satisfies the following condition:

1. Bigger than zero, the program will exit when n is zero.
2. Indivisible by 7.
3. Less than or equal to n.

(* The input n is at most 10000)

Then sum them up!

Input:
5
10
15
0

Output:
1 + 2 + 3 + 4 + 5 = 15
1 + 2 + 3 + 4 + 5 + 6 + 8 + 9 + 10 = 48
1 + 2 + 3 + 4 + 5 + 6 + 8 + 9 + 10 + 11 + 12 + 13 + 15 = 99

Gist: https://github.com/benchen216/NCTU-GoProgramming-2021

Hint

fmt.Println()
fmt.Print()
fmt.Printf()

https://golang.org/pkg/fmt/#Print