LECTURE 4.2: CONTROL FLOW / IF STATEMENT

BY LINA HAMMAD & AHMAD BARGHASH

In this lecture, you will learn to use if expression in Kotlin with the help of examples.

FLOW CONTROL

- In any programming language, we need **control statements** to control the flow of the program based on the output of a condition. For example, if a number is even then display "even number" but if the number is odd then display "odd number". To achieve this in programming, we need to use control statements that check whether a condition is satisfied or not, if it does then do this, if not then skip this. In **kotlin we use expressions to control flow** in the program.

FLOW CONTROL

- In any programming language, we need **control statements** to control the flow of the program based on the output of a condition. For example, if a number is even then display "even number" but if the number is odd then display "odd number". To achieve this in programming, we need to use control statements that check whether a condition is satisfied or not, if it does then do this, if not then skip this. In **kotlin we use expressions to control flow** in the program.

- There are several types of **expressions used in Kotlin:**

1. If expression
2. If..else expression
3. If..else if..else expression
4. Nested expressions
5. When
6. For loop
7. While loop and do while loop

IF EXPRESSION

- Use the if statement to specify a block of kotlin code to be executed if a **condition is true**.
- In Kotlin **if** is an expression, it is not a keyword. The expression **"if"** will return a value whenever necessary.
- The **if** expression is used to create simple conditional tests. It can be used in conjunction with the else keyword.
- Like other programming language, **"if-else"** block is used as an initial conditional checking operator. In the following example, we will compare two variables and provide the required output accordingly.

IF EXPRESSION

- The syntax of if is:

```
if (condition) {
    // block of code to be executed if the condition is true
}
```

- Here we have a condition in the if expression, if the condition returns true then the statements inside the body of if expression are executed, if the condition returns false then they are completely ignored.
- In other words, If..Else expression is used when we need to perform some action if a condition is true and we need to perform a different action when a condition is false.
- The curly braces are optional if the body of if has only one statement.

IF EXPRESSION EXAMPLE

- In this example, if the given number is even then we are displaying "Even Number" in the output, else we are skipping the statements inside "if".

```
fun main() {

    val number = 100

    // if expression
    if (number%2 == 0) {
        println("Even Number")
    }

    println("Out Of If statement")
}
```

The output is:

```
Even Number
Out Of If statement
```

```
fun main() {

    val number = 101

    if (number%2 == 0){
        println("Even Number")
    }
    println("Out Of If statement")
}
```

The output is:

```
Out Of If statement
```

IF - ELSE EXPRESSION

- The syntax of if - else is:

```
if (testExpression) {
    // codes to run if testExpression is true
}
else {
    // codes to run if testExpression is false
}
```

- if executes a certain section of code if the testExpression is evaluated to true. It can have optional else clause. Codes inside else clause are executed if the testExpression is false.

TRADITIONAL USAGE OF IF...ELSE

```
fun main() {
    val number = -10
    if (number > 0) {
        print("Positive number")
    } else {
        print("Negative number")
    }
}
```

The output is:

```
Negative number
```

```
fun main() {
    val number = 12
    if (number > 0) {
        print("Positive number")
    } else {
        print("Negative number")
    }
}
```

The output is:

```
Positive number
```

TRADITIONAL USAGE OF IF...ELSE

```
fun main() {

    var a = 10
    var b = 15
    var max:Int
    if (a < b) {
        max = b
    }else{
        max = a

    }
    println ("The maximum number is $max")
}
```

The output is:

```
The maximum number is 15
```

TRADITIONAL USAGE OF IF...ELSE

```
fun main() {
    // Marks out of 100
    val marks = 90
    if (marks < 30) {
        println("Father will get angry")
    }
    else {
        println("Father will give me money")
    }
}
```

The output is:

```
Father will give me money
```

IF AS EXPRESSION IN KOTLIN

- Unlike Java (and other many programming languages), if can be used an expression in Kotlin; it returns a value. For example:

```
fun main() {
    val number = -10
    val result = if (number > 0) {
        "Positive number"
    } else {
        "Negative number"
    }
    println(result)
}
```

The output is:

```
Negative number
```

- **Note**: The else branch is mandatory when using if as an expression.

IF AS EXPRESSION IN KOTLIN

- **Remember**: the curly braces are optional if the body of if has only one statement.

```
fun main( ) {
    val number = -10
    val result = if (number > 0) "Positive number" else "Negative number"
    println(result)
}
```

The output is:

```
Negative number
```

IF AS EXPRESSION IN KOTLIN

- If the block of if branch contains more than one expression, the last expression is returned as the value of the block.

```
fun main( ) {
    val a = -9
    val b = -11
    val max = if (a > b) {
        println("$a is larger than $b.")
        println("max variable holds value of a.")
        a
    } else {
        println("$b is larger than $a.")
        println("max variable holds value of b.")
        b
    }
    println("max = $max")
}
```

The output is:

```
-9 is larger than -11.
max variable holds value of a.
max = -9
```

IF..ELSE IF..ELSE LADDER EXPRESSION

- In this expression we have one “if” block, one “else” block and one or more “else if” blocks. This is used for checking multiple conditions.
- The curly braces are optional if the body of if has only one statement.

IF..ELSE IF..ELSE EXAMPLE

- In this example, we have a number and we are checking whether it's a negative number, single digit number, two-digit number or multiple digit number. We are checking these multiple conditions using if..else if..else expression. When none of the condition returns true then the statements inside the "else" block gets executed.

```
fun main() {
    val num = 99
    if(num < 0)
        println("Number is Negative")
    else if (num > 0 && num < 10)
        println("Single digit number")
    else if (num >= 10 && num < 100)
        println("Double digit number")
    else
        println("Number has 3 or more digits")
}
```

The output is:

```
Double digit number
```

IF..ELSE IF..ELSE EXAMPLE

- This program checks whether number is positive number, negative number, or zero.

```
fun main() {
    val number = 0
    val result = if (number > 0)
        "positive number"
    else if (number < 0)
        "negative number"
    else
        "zero"

    println("number is $result")
}
```

The output is:

```
number is zero
```

NESTED IF EXPRESSION

- When one expression is present inside another expression body then it is called nesting of expressions. For example if an "if expression" is present inside another "if" then it is called "nested if" expression.
- The syntax of nested if:

```
if (testExpression) {
    // codes to run if 1 testExpression is true
}else {
    // codes to run if 1 testExpression is false
    if (testExpression) {
        // codes to run if 2 testExpression is true
    }else {
        // codes to run if 2 testExpression is false
    }
}
```

NESTED IF EXPRESSION EXAMPLE

```
fun main() {
    val num = 101
    if(num<0)
        println("Negative Number")
    else {
        //Nested expression
        if(num%2 == 0)
            println("Even Number")
        else
            println("Odd Number")
    }

}
```

The output is:

```
Odd Number
```

NESTED IF EXPRESSION EXAMPLE

- This program computes the largest number among three numbers.

```
fun main() {
    val n1 = 3
    val n2 = 5
    val n3 = -2
    val max = if (n1 > n2) {
        if (n1 > n3)
            n1
        else
            n3
    } else {
        if (n2 > n3)
            n2
        else
            n3
    }
    println("max = $max")
}
```

The output is:

```
max = 5
```

ASSIGNMENT 1

- Write a program that declares and 5 integer variables that correspond to 5 grades and give them random variables of your choice. The grades are in the scale 0-25 where 25 is the maximum point. Your program should perform the following
  - Covert all grades to the 100% scale and save them in the same variables. Hint multiply by 4
  - Calculate and print how many failed classes the student have. Hint grade<50%
  - Calculate and print the sum and the average of all grades
  - Classify the average. In case student got average >50 and <75 print Good overall. In case student got average >=75 and <85 print Very Good. In case student got average >=85 print Excellent.