INDEX :

Basics Of Kotlin :

```kotlin
fun main() {

    /*       **** Variables ****

    # var -> Variables that can be reassigned

    */

    // 1. Dynamic var

    var rollNo = 12

    var mobileNumber = 6306805525L

    var price = 80.50f

    var totalCost = 665.6486846

    var gender = 'M'

    var name = "Its Me"

    var isActive = true


    println(rollNo)


    // Reassiging the value ->

    rollNo = 21

    println(rollNo)


    // 2. Specific Datatype

    var myRoll : Int = 123 // -> here we specify that var must be 'Int' variable type.

    println(myRoll)

    // Reassiging the value ->

    myRoll = 321

    println(myRoll)


    /*

    # val -> Read only local variables are defined using the keyword val. They can be assigned

    only once.

    */

    // 1. Dynamic val

    val dyn = 12

    // dyn = 21   // -> this will give error.

    println(dyn)
```

```
    // 2. Specific Datatype

    val spc : Int = 21

    println(spc)


    // To print the value of myRoll we can write code as follow method :-

    println("The value of myRoll is : " + myRoll)

    // Or we can also write it as follow :-

    println("The value of myRoll is : $myRoll")


    // To add two numbers we need to do so :-

    val a = 12

    val b = 21

    println("Addition of the numbers is :  " + (a+b))

    println("Value of a is :  $a")

    println("Value of b is :  $b")

    println("Addition of the number is :  ${a+b}")



}
```

Output :-

```
A_BasicsOfKotlin.kt:12:9: warning: variable 'isActive' is never used
    var isActive = true
        ^
12
21
123
321
12
21
The value of myRoll is : 321
The value of myRoll is : 321
Addition of the numbers is :  33
Value of a is :  12
Value of b is :  21
Addition of the number is :  33
PS D:\15. Tutorial Of Kotlin>
```

Operators :

```
fun main() {

    // **** Arithmetical Operators ****

    val a = 21

    val n = 12

    val add = a + n

    println(add)

    var sub = a - n

    println(sub)

    val multi = a * n

    println(multi)

    val div = a / n

    println(div)

    val mod = a % n

    println(mod)


    println(n..a)

    // (n..a) -> this will give the range from 'n' to 'a'. We need to use for loop to display range.

    for(i in n..a){

        println(i)

    }


    // **** Comparison Operators ****

    println(a > n)

    println(a < n)

    println(a >= n)

    println(a <= n)


    // **** Equality Operators and Inequality Operators ****

    println(a == n)

    println(a != n)


}
```

Output :-

```
PS D:\15. Tutorial Of Kotlin> cd
33
9
252
1
9
12..21
12
13
14
15
16
17
18
19
20
21
true
false
true
false
false
true
PS D:\15. Tutorial Of Kotlin>
```

User Input :

```
import java.util.*


fun main() {

    // User 'Input' using readLine()


    print("Enter a variable here :  ")

    val name = readLine()   // Dynamic

    println("Entered variable is :  $name")


    print("Enter a string here :  ")

    val myName : String? = readLine()   // Specified

    println("Entered string is :  $myName")


    println(myName!!::class.simpleName)  // -> will give the datatype of myName


    // Following is the way for type-casting

    println("Enter the roll number :  ")

    val roll = readLine()!!.toInt()

    println("Entered roll number is :  $roll")

    println(roll::class.simpleName)


    println("Enter the float value :  ")

    val fee = readLine()!!.toFloat()

    println("Entered fee is :  $fee")

    println(fee::class.simpleName)


    println("Taking input from user by using Scanner class")

    print("Enter some variable here :  ")

    val sc = Scanner(System.`in`)

    val varib = sc.next()

    println("Entered value is :  $varib")

    println("Entered variable formate is :  " + varib!!::class.simpleName)


    // For other data type
```

```
    print("Enter your age here :  ")

    val age = sc.nextInt();

    println("Entered age is :  $age")


}
```

Output :-

```
PS D:\15. Tutorial Of Kotlin> cd "d:\15. Tutorial

Enter a variable here :  123
Entered variable is :  123
Enter a string here :  Aman Verma
Entered string is :  Aman Verma
String
Enter the roll number :
21
Entered roll number is :  21
Int
Enter the float value :
21.12
Entered fee is :  21.12
Float
Taking input from user by using Scanner class
Enter some variable here :  1234
Entered value is :  1234
Entered variable formate is :  String
Enter your age here :  20
Entered age is :  20
PS D:\15. Tutorial Of Kotlin>
```

String :

```
fun main() {

    /*                                **** String ****

    String -> String are immutable. Once you initialize a string, you cann't change its value or

    assign a new value to it. All operations that transform string return their result in a new

    String object, leaving the original string unchanged.

    */


    val str = "Hello I'm here"

    println(str)

    for(i in str){

        println(i)

    }


    // **** Raw String **** -> This will print the string as you write in code.

    // Escaped String doesn't work in 'Raw' string

    val massage = """To,

    The principal

    College Of Vocational Studies


    Sir,

    I want to say that our college is not well as of others, so do something over this.

    'Escaped' string like \n, \t doesn't work in 'Raw' string.

    Thanking you!"""

    println(massage)


    val str1 = "It's Me"

    println("Given str1 is :  $str1")

    println("Given str1 in upper case is :  ${str1.uppercase()}") // -> way to use function.

    println("Given str1 in lower case is :  ${str1.lowercase()}")


}
```

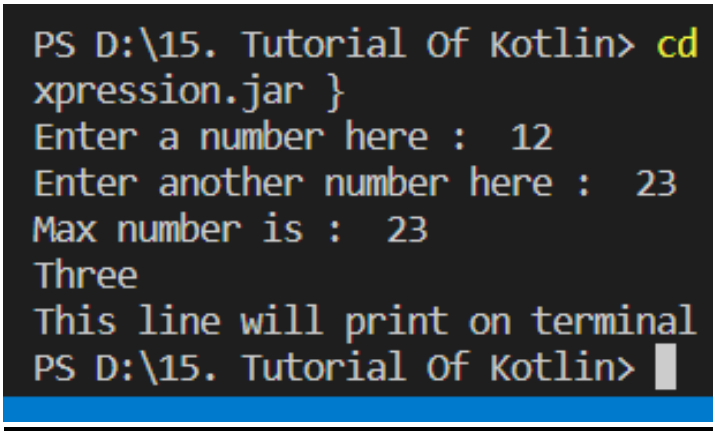
Output :-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

PS D:\15. Tutorial Of Kotlin> cd "d:\15. Tutorial Of Kotlin\" ; if ($?) { kotlinc D_String.kt
Hello I'm here
H
e
l
l
o

I
'
m

h
e
r
e
To,
    The principal
    College Of Vocational Studies

    Sir,
    I want to say that our college is not well as of others, so do something over this.
    'Escaped' string like \n, \t doesn't work in 'Raw' string.
    Thanking you!
Given str1 is :  It's Me
Given str1 in upper case is :  IT'S ME
Given str1 in lower case is :  it's me
```

When Expression :

```
fun main() {

    //            **** if Statement ****

    print("Enter a number here :  ")

    val num1 = readLine()!!.toInt()

    print("Enter another number here :  ")

    val num2 = readLine()!!.toInt()

    var max : Int

    max = if(num1>num2){

        num1

    }

    else{

        num2

    }

    println("Max number is :  $max")


    //              **** When Expression ****

    val x = 3

    when(x){    // Its act like 'switch' statment of C++ or Java

        1 -> println("One")

        2 -> println("Two")

        3 -> {

            println("Three")

            println("This line will print on terminal")

        }

        4, 5 -> println("Four or Five")  // If we need to display same thing with two different

        // variable, then we can write as so.

        // 5 -> println("Five")

        in 6..9 -> println("This will print in range of 6 to 9")


        else -> println("Not Valid !")

    }


}
```

Output :-


PS D:\15. Tutorial Of Kotlin> cd
xpression.jar }
Enter a number here : 12
Enter another number here : 23
Max number is : 23
Three
This line will print on terminal
PS D:\15. Tutorial Of Kotlin>

Loops :

```
fun main() {

    //            **** for loop ****

    // Syntax :- for( item in collection )

    for( i in 1..5){

        print("This is ")

        println("step :  $i")

    }


    // 'for' loop for reverse order

    println("'for' loop for reverse order")

    for(item in 5 downTo 1)

    println("This is step :  $item")


    //            **** while loop ****

    var x = 0

    while(x < 5){

        println("Current value of x is :  $x")

        x++

    }


    // println("Printing the value in the for loop :")

    // print("Enter the limit here :  ")

    // val init = readLine()!!.toInt()


    // print("Enter a number here :  ")

    // val num = readLine()!!.toInt()


    // println("Use of for loop for printing the table :-")

    // for(i in 1..init){

    //     println("$i X $num =  ${i*num}")

    // }



}
```

Output :-

```
PS D:\15. Tutorial Of Kotlin> cd "d:\15. Tutorial
This is step :  1
This is step :  2
This is step :  3
This is step :  4
This is step :  5
'for' loop for reverse order
This is step :  5
This is step :  4
This is step :  3
This is step :  2
This is step :  1
Current value of x is :  0
Current value of x is :  1
Current value of x is :  2
Current value of x is :  3
Current value of x is :  4
PS D:\15. Tutorial Of Kotlin>
```