Kotlin

Mail= real_time_programmer03@gmail.com (message if have any query)

Contents :-

➢ History of kotlin

a) Introduced in “May 2017” by Google and given by JITBRAINS company , which gives one of the most famous IDEA as INTELLIJ IDEA
b) Use to develop android application
c) Kotlin is modern programming language then java .
d) Its having more feature than java , internally it runs on the jvm then kotlin also got the features of jvm (features like the robust , security , platform independent , distributed , dynamic, portable, high performance)
e) We run kotilin code as desktop also as web application (JVM or As java script)

➢ Features of kotlin

a) No need of semicolon After the sentence completed .
b) There is no checked exceptions , it have only run time exception
c) In java there is many chances of occurring null pointer but in kotlin There is less case to occur the null pointer exception hence it also called as null safe language .
d) We can internally exchange or fit the kotlin code in java code .
e) It does not have new keyword .
f) There is no primitive data type, it use objects or wrapper class .
g) The object is created just calling the constructor as like other method.
h) Kotlin support (limited) operator overloading .
i) Kotlin is fully compatible with java 6 and next versions .
j) In java we use Scanner scan=new Scanner(System.in) but in kotlin we use “readLine()!!” Method.

➢ Code for printing statement in kotlin

Package

fun main(args:Array<String>)=println("kotlin example")

➢ Data type in kotlin

a) Kotlin doesn't support the primitive data type it use wrapper class or objects

b) Kotlin doesn't have an void key word . if we don't specify the return type to method it consider as the void . we also use Unit keyword instead of void.

c) Two types of variables in kotlin

1) **Mutable**

:- declare as var .

Value of that variable can change

Example => var today:Int = 10

var month: String= null //X

var month: String?= null // correct (? it mens I specify value later)

var name =" "

Name = readLine()!! (!! Is called assart not null operator, we telling the no the method is initialize , initially value is null but we initialize)

2) **Immutable**

:- declare as val .

Value of that variable cannot be change

Example => val id: Int = 10

d) We use + and $ to print original value .

Example : println("person name:"+name)

println("person name: $name")

➢ Array and collection

a) In Kotlin array are declaration and initialize as like follows

```
Var name:Array<string>= arrayOf(values1,values2,…)
Var name:IntArray= IntArrayOf(values1,values2,…)
```

b) For individually read array we use for loop like =

For(day in days)

Example =

```
Var days :Array<string>= arrayOf("sun","mon","tus","wednesday")
For( day in days)
{
Println(day)
}
```

c) For list and set type collection use as follows =

- Only difference between list and set is list allows duplicates and set doesn't allow duplicate.
- var list:List<String>=listOf("sun","mon","tus","wednesday")

```
for (x in list){
println(x)          }
```

But it is immutable if you want mutable we use as

```
var list:mutableList<String>=mutableLidtOf<String>()
list.add(“sun”)
for (x in list){
println(x)
}
```

➢ When and Range method

a) Instead of **switch** statement in java we use we use a **when** in kotlin

Example : var index=readLine()!!.toInt()

```
when(index){
        1-> println(“monday”)
        2-> println(“tusday”)
        3-> println(“wednesday”)
Else->println(“plz enter between 1-3”)
}
```

b) Range in kotlin is a range between two specific number like 1 to 100.

The range is shown by **..**

Example: for(x in 1..100){

```
println(x)
}
```

```
Output  1
        2
        3 upto
        100
```

We also specify the step to the range . step means the number jump by that numbers like

Example :

```
for(x in 1..100 step 2){
println(x)
}
Output  1
        3
        5
        7 upto
        100
```

➢ Constructor in Kotlin

a) Kotlin have two types of constructor
   1) Primary Constructor
   2) Secondary constructor
   - In kotlin for every class we can create only one Primary constructor any number of secondary constructore as per requirement
   - Primary constructor means the constructor which have arguments followed by class name
   - Secondary constructor can be given by "constructor()", if we required to specify argument then specify in the braces .
   - Init use to specify the return value to the varable

   Example :

```
fun main(args:Array<string>){
val i= Info(year:2008)// value for primary constructor
Info(year:"two thousand eight")// value for secondary constructor
}
class Info(year:Int) {// class with primary constructor
var current_year:Int?=nul
init{
current_year= year
}
println(current_year)
construtor(year:String){// secondary string
println(year)
}
}
```

```
Output =  2008
          two thousand eight
```

➢ extending class and interface in Kotlin

- we can extand class and interface in kotlin like java but java use the extend or implement keyword for that but kotlin use **:** for that .
- for the extending the class the parent class should be the open for the extension .for that use open key word like we use private , protected, public keyword in java .
- In kotlin for extending class we can extends the that class constructor .

  Example:

```
fun main(args:Array<String>){
var b = Bus()
b.travel()
b. noof_wheels=4
}
open class Vehical{
var noof_wheels:Int? = null
fun travel(){
println("  travelling started ")
}
}
Interface Insurance{
fun Insuranceofvehical()
}
class Bus: Vehical(),Insurance {// here we use Vehical class constructor
                                    for the extend class
override fun Insuranceofvehical() {
 }
 }

Output =  travelling started
```

➢ Function in Kotlin

- We can declare function in kotlin using fun keyword
- There are two types of function inn kotlin

1) Inline function
2) Multiline function

1) Inline function

- Inline function consist the all code inn one line
- Example

```
fun main(args:Array<String>){
val num1= readLine()!!.toInt()
val num2= readLine()!!.toInt()
println(sum(num1,num2))
}
fun sum(a:Int,b:Int)=a+b// output = 10  20 = 30
fun displayMessage() = println("welcome") // welcome
```

2) Multiline function

- This is like normal function
- Example

```
fun main(args:Array<String>){
displayMonthInfo()
}
fun displayMonthInfo(){
println("this is 27 june")
println("two thousand eighteen ")
}
Output=
          this is 27 june
          two thousand eighteen
```