Practical No.: 01

i) Write a program using Kotlin to implement control structures and loops.

Control Structures:

a) if…else

```
fun main(args: Array<String>) {
    val age:Int = 20
    val result = if (age > 12) {
       if ( age > 12 && age  < 20 ){
           "Teen"
       }else{
           "Adult"
       }
    } else {
        "Minor"
    }
    print("The value of result : ")
    println(result)
}
```

b) when

```
fun main(args: Array<String>) {
```

```
val day = 2

when (day) {
  1 -> {
     println("First day of the week")
     println("Monday")
  }
  2 -> {
     println("Second day of the week")
     println("Tuesday")
  }
  3 -> {
     println("Third day of the week")
     println("Wednesday")
  }
  4 -> println("Thursday")
  5 -> println("Friday")
  6 -> println("Saturday")
  7 -> println("Sunday")
  else -> println("Invalid day.")
}
```

```
}
```

Loops:

a) for loop

```
fun main(args: Array<String>) {
   var fruits = arrayOf("Orange", "Apple", "Mango", "Banana")
   for (index in fruits.indices) {
      println(fruits[index])
   }
}
```

b) while loop

```
fun main(args: Array<String>) {
   var i = 5;
   while (i > 0) {
      println(i)
      i--
   }
}
```

c) do…while loop

```
fun main(args: Array<String>) {
   var i = 5;
```

```
    do{
        println(i)
        i--
    }while(i > 0)
}
```

ii) Write a program to implement object-oriented concepts in Kotlin.

a) Class and Objects

```
class myClass {
    // Property (data member)
    private var name: String = "Udemy"

    // Member function
    fun printMe() {
        print("The best Online Education Sites - " + name)
    }
}
fun main(args: Array<String>) {
    val obj = myClass() // Create object obj of myClass class
    obj.printMe() // Call a member function using object
}
```

b) Constructors

```
class Person{
   // Member Variables
   var name: String
   var age: Int
   var salary:Double

   // First Secondary Constructor
   constructor ( _name: String, _age: Int) {
      this.name = _name
      this.age = _age
      this.salary = 0.00
      println("Name = $name")
      println("Age = $age")
   }

   // Second Secondary Constructor
   constructor ( _name: String, _age: Int, _salary: Double) {
      this.name = _name
      this.age = _age
```

```
        this.salary = _salary
        println("Name = $name")
        println("Age = $age")
        println("Salary = $salary")
    }
}

fun main(args: Array<String>) {
    val nuha = Person("Nuha", 12)
    val zara = Person("Zara", 20, 2000.00)
}
```

c) Inheritance

```
open class ABC {
    fun think () {
        println("Hey!! I am thiking ")
    }
}
class BCD: ABC(){ // inheritence happend using default constructor
}
```

```
fun main(args: Array<String>) {

   var  a = BCD()

   a.think()

}
```