Kotlin is statically typed programming language

1. Variables
    a. `val name = "Bill Gates" // Val cannot be reassigned`
    b. `var country = "USA"  // value can be changed`
    c. Type Interface
        i. `val greeting = "Hello, World"  // type inferred as `String``
        ii. `val year = 2018  // type inferred as `Int``
    d. Type Interface
        i. `val greeting = "Hello, World"  // type inferred as `String``
        ii. `val year = 2018`
    e. Explicitly defining the type of variables
        i. `val greeting: String = "Hello, World"`
        ii. `val year: Int = 2018`
2. Conversion
    a. `val myInt = 100  => val myLong = myInt.toLong()  // 'Int' to 'Long'`
    b. `val doubleValue = 176.80 => val intValue = doubleValue.toInt()  // 176`
    c. `val myInt = 1000 => myInt.toString()  // "1000"`
    d. `val str = "1000" => val intValue = str.toInt()`
    e. `val str = "1000ABC" => str.toInt()  // Throws java.lang.NumberFormatException`
3. Operator
    a. `var a = 10, var b = 20`
    b. `var c = ((a + b) * ( a + b))/2  // 450`
4. When
    a. `var dayOfWeek = 4`
    b. `when(dayOfWeek) {`
    c. `1 -> println("Monday")`
5. Arrays
    a. `var numbers = arrayOf(1, 2, 3, 4, 5)`
    b. `var animals = arrayOf("Cat", "Dog", "Lion", "Tiger")`
    c. `var mixedArray = arrayOf(1, true, 3, "Hello", 'A')  // Works and creates an array of Objects`
    d. `var numArray = arrayOf<Int>(1, 2, 3, 4, "Hello")  // Compiler Error`
    e. `val myDoubleArray = arrayOf(4.0, 6.9, 1.7, 12.3, 5.4)`
    f. `val firstElement = myDoubleArray[0]`
    g. `val lastElement = myDoubleArray[myDoubleArray.size - 1]`

```
    h. val myCharArray = charArrayOf('K', 'O', 'T')  // CharArray(Java 'char[]')
    i. val myIntArray = intArrayOf(1, 3, 5, 7)// IntArray (corresponds to Java 'int[]')
    j. var mySquareArray = Array(5, {i -> i * i})     // [0, 1, 4, 9, 16]
6. Do while loop
    a. var x = 1
    b. do {
    c.      print("$x ")
    d.      x++
    e. } while(x <= 5)
7. For Loop
    a. for(value in 1..10) {
    b.      print("$value ")
    c. }
8. For Loop
    a. var primeNumbers = intArrayOf(2, 3, 5, 7, 11)
    b.
    c. for(number in primeNumbers) {
    d.      print("$number ")
    e. }
9. Indices
    a. var primeNumbers = intArrayOf(2, 3, 5, 7, 11)
    b. for(index in primeNumbers.indices) {
    c.      println("PrimeNumber(${index+1}): ${primeNumbers[index]}")
    d. }
10. Iteration using withIndex()
    a. var primeNumbers = intArrayOf(2, 3, 5, 7, 11)
    b. for((index, number) in primeNumbers.withIndex()) {
    c.      println("PrimeNumber(${index+1}): $number")
    d. }
11. val map = HashMap<String, String>() map["asd"] = "s" map.put("34", "354")
12.
13. Function
    a. fun avg(a: Double, b: Double): Double {   // Return Double type
    b.      return  (a + b)/2
    c. }
```

14. Model Class

    a. val steveJobs= User("Steve Jobs", 56)

    b. val (name, age) = steveJobs

    c. steveJobs.component1() // name

    d. steveJobs.component2() // age

Class Declaration:

```
data class User(val name: String, val email: String, val country: String)
```

Link: https://www.callicoder.com/kotlin-control-flow/