**502-02 : Advance Mobile Computing**

**Unit-1: Introduction to Kotlin**

**1.1 Concepts of Kotlin and its introduction.**

Kotlin Tutorial

Kotlin tutorial provides basic and advanced concepts of Kotlin programming language. Our Kotlin tutorial is designed for beginners and professionals both.

Kotlin is a statically-typed, general-purpose programming language. It is widely used to develop android applications.

Our Kotlin Tutorial includes all topics of Kotlin such as introduction, architecture, class, object, inheritance, interface, generics, delegation, functions, mixing Java and Kotlin, Java vs. Kotlin, etc.

**What is Kotlin**

**Kotlin** is a general-purpose, statically typed, and open-source programming language. It runs on JVM and can be used anywhere Java is used today. It can be used to develop Android apps, server-side apps and much more.

History of Kotlin

**Kotlin** was developed by JetBrains team. A project was started in 2010 to develop the language and officially, first released in February 2016. Kotlin was developed under the Apache 2.0 license.

Features of Kotlin

* **Concise:**Kotlin reduces writing the extra codes. This makes Kotlin more concise.
* **Null safety:**Kotlin is null safety language. Kotlin aimed to eliminate the NullPointerException (null reference) from the code.Interoperable.
* **Interoperable:**Kotlin easily calls the Java code in a natural way as well as Kotlin code can be used by Java.
* **Smart cast:**It explicitly typecasts the immutable values and inserts the value in its safe cast automatically.
* **Compilation Time:**It has better performance and fast compilation time.
* **Tool-friendly:**Kotlin programs are build using the command line as well as any of Java IDE.
* **Extension function:**Kotlin supports extension functions and extension properties which means it helps to extend the functionality of classes without touching their code.

1.2 Downloading IntelliJ and its settings.

1.3 Variables:

1.3.1 val vs. var, Byte, Short, Int, Long, Float, Double, Boolean, and Char.

**Kotlin Variable**

Variable refers to a memory location. It is used to store data. The data of variable can be changed and reused depending on condition or on information passed to the program.

**Variable Declaration**

Kotlin variable is declared using keyword **var** and **val**.

1. var language ="Java"
2. val salary = 30000

The difference between var and val is specified later on this page.

Here, variable language is String type and variable salary is Int type. We don't require specifying the type of variable explicitly. Kotlin complier knows this by initilizer expression ("Java" is a String and 30000 is an Int value). This is called type inference in programming.

We can also explicitly specify the type of variable while declaring it.

1. var language: String ="Java"
2. val salary: Int = 30000

It is not necessary to initialize variable at the time of its declaration. Variable can be initialized later on when the program is executed.

1. var language: String
2. ... ... ...
3. language = "Java"
4. val salary: Int
5. ... ... ...
6. salary = 30000

Difference between var and val

* **var** (Mutable variable): We can change the value of variable declared using **var** keyword later in the program.
* **val** (Immutable variable): We cannot change the value of variable which is declared using **val** keyword.

**Example**

1. var salary = 30000
2. salary = 40000 //execute

Here, the value of variable salary can be changed (from 30000 to 40000) because variable salary is declared using **var** keyword.

1. val language = "Java"
2. language = "Kotlin" //Error

Here, we cannot re-assign the variable language from "Java" to "Kotlin" because the variable is declared using **val** keyword.

**Kotlin Data Type**

**Data type** (basic type) refers to type and size of data associated with variables and functions. Data type is used for declaration of memory location of variable which determines the features of data.

In Kotlin, everything is an object, which means we can call member function and properties on any variable.

Kotlin built in data type are categorized as following different categories:

* Number
* Character
* Boolean
* Array
* String

Number Types

Number types of data are those which hold only number type data variables. It is further categorized into different Integer and Floating point.

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Bit Width (Size)** | **Data Range** |
| Byte | 8 bit | -128 to 127 |
| Short | 16 bit | -32768 to 32767 |
| Int | 32 bit | -2,147,483,648 to 2,147,483,647 |
| Long | 64 bit | -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 |
| Float | 32 bit | 1.40129846432481707e-45 to 3.40282346638528860e+38 |
| Double | 64 bit | 4.94065645841246544e-324 to 1.79769313486231570e+308 |

Character (Char) Data Type

Characters are represented using the keyword **Char**. Char types are declared using single quotes ('').

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Bit Width (Size)** | **Data Range** |
| Char | 4 bit | -128 to 127 |

**Example**

1. val value1 = 'A'
2. //or
3. val  value2: Char
4. value2= 'A'

Boolean Data Types

Boolean data is represented using the type **Boolean**. It contains values either true or false.

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Bit Width (Size)** | **Data Value** |
| Boolean | 1 bit | true or false |

**Example**

1. val flag = **true**

Array

Arrays in Kotlin are represented by the Array class. Arrays are created using library function arrayOf() and Array() constructor. Array has get (), set() function, size property as well as some other useful member functions.

Creating Array using library function arrayOf()

The arrayOf() function creates array of wrapper types. The item value are passed inside arrayOf() function like arrayOf(1,2,3) which creates an array[1,2,3].

The elements of array are accessed through their index values (array[index]). Array index are start from zero.

1. val id = arrayOf(1,2,3,4,5)
2. val firstId = id[0]
3. val lasted = id[id.size-1]

Creating Array using Array() constructor

Creating array using Array() constructor takes two arguments in Array() constructor:

1. First argument as a size of array, and
2. Second argument as the function, which is used to initialize and return the value of array element given its index.
3. val asc = Array(5, { i -> i \* 2 }) //asc[0,2,4,6,8]

String

String in Kotlin is represented by String class. String is immutable, which means we cannot change the elements in String.

**String declaration:**

1. val text ="Hello, JavaTpoint"

Types of String

String are categorize into two types. These are:

1. **Escaped String:**Escape String is declared within double quote (" ") and may contain escape characters like '\n', '\t', '\b' etc.

1. val text1 ="Hello, JavaTpoint"
2. //or
3. val text2 ="Hello, JavaTpoint\n"
4. //or
5. val text3 ="Hello, \nJavaTpoint"

2. **Raw String:**Row String is declared within triple quote (""" """). It provides facility to declare String in new lines and contain multiple lines. Row String cannot contain any escape character.

1. val text1 ="""
2. Welcome
3. To
4. JavaTpoint
5. """

1.3.2 String, Nullable variables.

1.4 Conditional statements: if and when. Difference between if and when.

1.4.1 ranges, types, values of function calls

1.5 Arrays and Lists:

1.5.1 create, modify, and access arrays

1.5.2 creating, modifying, and accessing lists

1.6 Loops (Iterative statements)

1.6.1 for and while loop.

1.6.2 break, continue and return