**Java**

**2. What is JAVA?**

**--------------------------------------------------------------------------------------------------------------------------------------**

1. **History of JAVA:**

**--------------------------------------------------------------------------------------------------------------------------------------**

**10. Java Data Types**

**The various sizes and values that can be stored in a variable are referred to as data types.**

**Each type of data is predefined programming language.**

**These data types makes error free java code**

**There are two categories in Java where data types are divided:**

1. **Primitive Data Type**
2. **Non-Primitive Data Type**

**1. Primitive Data Type**

**Primitive data types are predefined function and has a keyword.**

**The fundamentals of manipulating data.**

**There 8 types of primitive data type:**

**1 byte = 8 bits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Size** | **Default** | **Description** | **Range of Values** |
| **boolean** | **1 bit** | **False** | **True or False** | **True,False** |
| **byte** | **1 byte** | **0** | **Two’s complement integer** | **-128 to 127** |
| **short** | **2 byte** | **0** | **Two’s complement integer** | **-32,768 to 32,767** |
| **long** | **8 byte** | **0** | **Two’s complement integer** | **-263 to 263 -1** |
| **int** | **4 byte** | **0** | **Two’s complement integer** | **-231 to 231 -1** |
| **float** | **4 byte** | **0.0** | **IEEE 754 floating point** | **Upto 7 decimal digits** |
| **double** | **8 byte** | **0.0** | **IEEE 754 floating point** | **Upto 15 decimal digits** |
| **char** | **2 byte** | **\u0000** | **Unicode character** | **Characters representation of ASCII values** |

**Boolean data type**

* **Boolean data type represents single bit information like true/false (or) yes/no.**
* **Mostly used for conditional testing**

**Syntax:**

**boolean booleanvar;**

**Example:**

**Program 1:**

**public class Boolean1 {**

**public static void main(String[] args) {**

**boolean  one = true;**

**boolean two = false;**

**System.out.println(one);**

**System.out.println(two);**

**}**

**}**

**Output:**

**True**

**False**

**Program 2:**

**public class Boolean2{**

**public static void main(string[] args){**

**int a=11;**

**int b=20;**

**boolean c=true;**

**boolean d=false;**

**if(a<b)**

**{**

**System.out.println(c);**

**}**

**else**

**{**

**System.out.println(d);**

**}**

**}**

**}**

**Output:**

**True**

**Byte data type**

* **The byte data type is used to save memory  in large arrays.**
* **Byte is 4 times smaller than an integer (int).**
* **8 bit of two’s complement integer.**

**Syntax :**

**byte num;**

**Example:**

**Program 1:**

**public class byte1{**

**public static void main(String[]args){**

**byte a=3;**

**System.out.println(a);**

**}**

**}**

**Output :**

**3**

**Program 2:**

**public class byte2{**

**public static void main(String[]args){**

**byte age1=18;**

**byte age2=45;**

**System.out.println("Father age:"+age2);**

**System.out.println("daugther age:"+age1);**

**}**

**}**

**Output :**

**Father age:45**

**daugther age:18**

**Short data type**

* **Short data type is also used to save memory like byte.**
* **Short is 2 times smaller than int.**
* **16 bit of two’s complement integer.**

**Syntax:**

**short val;**

**Example :**

**Program 1:**

**public class short1{**

**public static void main(String[]args){**

**short num = 2011;**

**System.out.println(num);**

**}**

**}**

**Output :**

**2011**

**Program 2:**

**public class short2{**

**public static void main(String[]args){**

**short num = 33456;**

**System.out.println(num);**

**}**

**}**

**Output :**

**1 error**

**Because its range is from -32768 to 32767.**

**Long data type**

* **Long data type is used when more the int provide.**
* **Its range values is large number od value.**

**Syntax:**

**long val;**

**Example**

**Program 1:**

**public class long1{**

**public static void main(String[]args){**

**long num = 699424789;**

**System.out.println(num);**

**}**

**}**

**Output :**

**699424789**

**Float data type**

* **Float data type used to store fractional values.**
* **End the value with”f”.**
* **The float data type is a single-precision 32-bit IEEE 754 floating point.**
* **To save memory in large arrays float data type is used.**

**Syntax :**

**float val;**

**Example :**

**Program 1:**

**public class float1{**

**public static void main(String[]args){**

**float num = 4.56f;**

**System.out.println(num);**

**}**

**}**

**Output :**

**4.56**

**Program  2:**

**public class float2{**

**public static void main(String[]args){**

**float num1 = 1.2345678f;**

**float num2 = 2.2345678f;**

**float result = num1 + num2;**

**System.out.println(result);**

**}**

**}**

**Output :**

**3.4691358**

**Double data type**

* **Double data type is also used to decimal values.**
* **The double data type is a double-precision 64-bit IEEE 754 floating point.**
* **End value with “d”.**

**Syntax :**

**double val;**

**Example :**

**Progarm 1:**

**public class double1{**

**public static void main(String[]args){**

**double num = 7.8945;**

**System.out.println(num);**

**}**

**}**

**Output :**

**7.8945**

**Program 2:**

**public class double2{**

**public static void main(String[]args){**

**double num1 = 7.8945;**

**double num2 = 6.8234;**

**double result = num1 + num2;**

**System.out.println(result);**

**}**

**}**

**Output :**

**14.7179**

**Char data type**

* **The char data type is used to store characters.**
* **Each character has its number based on the  reference of ASCII values.**
* **The char data type is a single 16-bit Unicode character.**

**Syntax :**

**char var;**

**Example :**

**Program 1:**

**public class char1{**

**public static void main(String[]args){**

**char apple = 'A';**

**System.out.println(apple);**

**}**

**}**

**Output :**

**A**

**Progarm 2:**

**public class char2{**

**public static void main(String[]args){**

**char a = 45;**

**char b = 56;**

**char E = 78;**

**System.out.println(a);**

**System.out.println(b);**

**System.out.println(E);**

**}**

**}**

**Output :**

**-**

**8**

**N**

**2. Non-Primitive Data Type**

* **Non-Primitive data type is also known as reference/object data type.**
* **The default value is null.**
* **The reference data type is generally used to declare to the representative refers of the object.**
* **Some of the example are string, array,classes,interface and object.**

**Classes and Objects**

* **Objects can be created using classes, which are user-defined data types.**
* **A class is made up of common attributes and methods shared by all of the class's objects.**

**Example :**

**public class Scooty{**

**String name;**

**double speed;**

**double mileage;**

**public void startScooty(){**

**System.out.println("Scooty Started");**

**}**

**public static void main(String[]args){**

**Scooty scooty1=new Scooty();**

**scooty1.name="activa";**

**Scooty scooty2=new Scooty();**

**scooty2.name="dio";**

**System.out.println(scooty2.name);**

**scooty2.startScooty();**

**System.out.println(scooty1.name);**

**}**

**}**

**Output :**

**dio**

**Scooty Started**

**activa**

**String**

* **Special function.**
* **Character arrays keep each character individually.**
* **Strings, on the other hand, can have character sequences within a single variable.**
* **Used to print a sentence not like character which print only single variable.**

**Syntax :**

**String <string\_variable\_name> = new String(“<sequence\_of\_characters>”);**

**Example :**

**public class Main {**

**public static void main(String[] args) {**

**String str = ("Life");**

**System.out.println("string is:"+ str);**

**String str1 = new String("long");**

**System.out.println( "String2 is:"+ str1);**

**}**

**}**

**Output :**

**string is:Life**

**String2 is:long**

**Array:**

* **Collection of data that are similar type.**
* **Java that are utilized to contiguously store pieces of the same data type.**
* **All of an array's items can be retrieved using a single reference name.**
* **The way that elements are saved is indexed, with index zero.**

**Syntax:**

**<data\_type> <array\_name> [] = {array\_item\_values};**

**Example :**

**public class array1{**

**public static void main(String[]args){**

**int[] studentID ={4,6,9,10};**

**System.out.println(studentID[2]);**

**System.out.println(studentID.length);**

**}**

**}**

**Output :**

**9**

**4**

**Interface :**

* **Implements is the keyword.**
* **Multiple interface can be created.**
* **Interface is not a class.**
* **Every method on interface is public abstract.**

**Example :**

**interface example {**

**void add();**

**void div();**

**}**

**class Solve implements example {**

**int a = 10, b = 20, c;**

**public void add() {**

**int c = a + b;**

**System.out.println("Addition of numbers: " + c);**

**}**

**public void div() {**

**int c = b / a;**

**System.out.println("Division of numbers: " + c);**

**}**

**}**

**public class Main {**

**public static void main(String[] args) {**

**Solve obj = new Solve();**

**obj.add();**

**obj.div();**

**}**

**}**

**Output :**

**Addition of numbers: 30**

**Division of numbers: 2**

**Needs for Data type:**

* **The data types are the in built function which is the fundamentals of the java language.**
* **These  reduces the errors in the program**
* **Data type specifies the size and type of values.**
* **Specify how memory stores the values of the variables.**
* **Each variables has different data type to store the variables.**

**Prerequisite:**

* **Types of list are there to store the variables respectively.**
* **Integers, string, character,boolean, etc, like this there are certain types is used to store.**

**--------------------------------------------------------------------------------------------------------------------------------------**

**14. If – else**

**--------------------------------------------------------------------------------------------------------------------------------------**

**15. Switch case:**

**16. For Loop:**

* **The number of repetition/iteration is fixed .**
* **We initialize a variable, check with the given condition and increment/decrement the value.**

**Syntax :**

* **for(initialization; condition; increment/decrement){**
* **// code to be executed**
* **}**

**Example :**

**Program :**

**public class loop1{**

**public static void main(String[] args){**

**int a=1;**

**for(a=1;a<=7;a++){**

**System.out.println(a);**

**}**

**}**

**}**

**Output :**

**1**

**2**

**3**

**4**

**5**

**6**

**7**