JAVA:::

1. JAVA installation—Download oracle java and install it.
2. Configure the PATH variable (C:\Program Files\Java\jdk-12.0.1\bin)
3. Type JAVA -VERSION to check if JAVA has been installed and configured properly.
4. Install Eclipse- IDE for writing JAVA code
5. Create a new Java project and class for writing the first program.
6. Datatypes—Int, bool, float, double, String
7. Java Operators: Arithmetic(+,-,\*,/,%), Relational(<,>,<=,>=,!=,==)—Returns Boolean values, Logical(&&,||,!), Assignment(=,+=,\*=,/=), Increment(++), Decrement(--)
8. Conditional Statements: If…If-Else…Nested If-Else if…Switch Case
9. Loops: while loop, do..while, for, for each,break,continue
10. Arrays: Datatype variable[]= new Datatype[Array size] or Datatype variable[]={value1,value2….}. (Object class is the parent datatype to hold any datatype values)---Single dimensional array
11. Multidimensional Array: int a[][]= new int[3][2];

Int a[][]={{1,2},{2,4},{2,3}}

1. String methods: length(),concat(),equals(),contains(),equalsignorecase(),substring(),replace()
2. OOPS concepts: Class, Objects and methods

Class—Logical entity; Collection of variables and methods.

Object—Physical entity. It is an instance of a class.

Method—It is a piece of code which performs certain tasks.

Classname obj= new Classname();

Classname()—instantiation

Obj---reference variable of object

Assign values to variables:

1. Using objects
2. Using constructors
3. Using methods

Constructor- It is a kind of method, having same name as the class. Does not return any value. It’s called at the time of object creation for initializing the values.

Methods- may or may not take parameters, may or may not return values.

1. Method overloading—Same method/constructor multiple times.

--Method signature should be different(number of parameters, different datatypes, order of parameters)

1. 2 types of constructor- Default and parameterized
2. This Keyword
3. Static keyword—Can be used for both variable and method. Does not require an object for accessing them. They are shared across all objects. Static methods can access static variables and methods. Static methods can also access non-static elements using objects. No-static methods can access everything without an object.
4. System.out.println()---

Class System{

Static PrintStream out;

}

1. JDK, JRE, JVM

JDK—Used for development using JAVA; contains JRE+JVM

JRE—Used for running JAVA applications

JVM—Responsible for executing JAVA programs. Compiles the programs. It looks for main method and executes.

1. Inheritance-Acquiring methods and variables from parent class to child class.

Class A{ //parent class//super class//base class

}

Class B extends A{// child class, sub class, derived class

}

Multi-level inheritance is possible but multiple inheritances cannot be done in Java.

1. Method overriding—Define the same method in parent class and child class
2. Final Keyword—Variables, method, Class. Variable value remains constant. Methods cannot be over-ridded. Class cannot be extended.
3. Interface—blueprint of a class. Contains only static and final variables. Methods are public by default and abstract methods(Only definition). We cannot instantiate an interface.

Interface A{

}

1. Public---accessible from everywhere

Default—accessible within the package

Protected— accessible within the package and outside but using inheritance

Private—Can be accessed only within the class

1. Exception Handling: Exception terminates the program unexpectedly.

2 types of exceptions—checked and unchecked. Checked exceptions are identified by java compiler. Unchecked are not identified by the compiler.

Handled by : Try-Catch block, Try-Catch-finally(always get executed), throws keyword—used at method level for checked exception.

1. ArrayList—Not fixed in size like array and can store multiple datatypes without using object datatype. ArrayList al= new ArrayList();
2. HashMap—key-value pair

For(Map.Entry m: hm.entryset()){m.getKey()+m.getValue()}