Day:2

Do in lab sessions

1. Java Modifiers.
   5. Java Modifiers.
   6. Access Modifiers.

Access Modifiers in Java are used to control the access or permission of a block of code. Its basically used for controlling the visibility or accessibility of a program/ block of code. It defines what parts of the program (such as classes, methods, constructors, etc.) will be accessible to the other member of the program.
Access Control comes along with the concept of **Encapsulation** in Object-Oriented Programming **OOPs in Java**.

Types of Access Modifiers in Java

- **Default Access Modifiers**
- **Private Access Modifiers**
- **Protected Access Modifiers**
- **Public Access Modifiers**

Summarized view of Accessibility of Access modifier in Java

Accessibility	**Default**	**Private**	**Protected**	**Default**
Inside Same Class	Yes	Yes	Yes	Yes
Subclass of a Class	Yes	No	Yes	Yes
Same Package Different Class	Yes	No	Yes	Yes
Different Package's Subclass	No	No	Yes	Yes
Different Package Non-Subclass	No	No	No	Yes

2. Java Buzzword.

1. Simple
2. Object-Oriented
3. Distributed
4. Compiled and Interpreted
5. Robust
6. Secure
7. Architecture-Neutral
8. Portable
9. High Performance
10. Multithreaded
11. Dynamic

3. Components of java

The Java architecture includes the three main components:

- Java Virtual Machine (JVM)
- Java Runtime Environment (JRE)
- Java Development Kit (JDK)

Java Virtual Machine

The main feature of Java is **WORA**. WORA stands for **Write Once Run Anywhere**. The feature states that we can write our code once and use it anywhere or on any operating system. Our Java program can run any of the platforms only because of the Java Virtual Machine. It is a Java platform component that gives us an environment to execute java programs. JVM's main task is to convert byte code into machine code.

JVM, first of all, loads the code into memory and verifies it. After that, it executes the code and provides a runtime environment

Java Runtime Environment

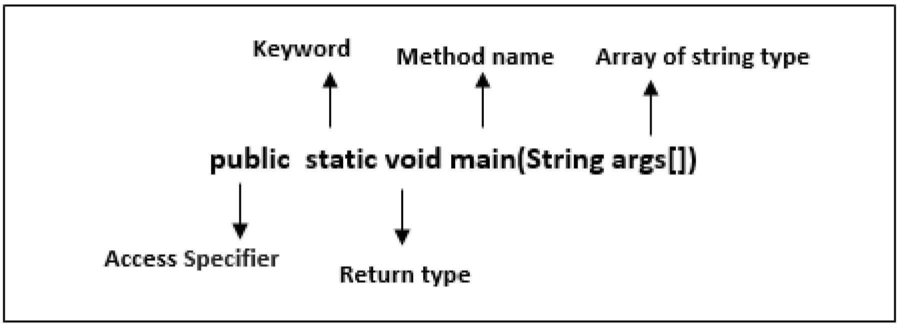
It provides an environment in which Java programs are executed. JRE takes our Java code, integrates it with the required libraries, and then starts the JVM to execute it.

Java Development Kit

It is a software development environment used in the development of Java applications and applets. Java Development Kit holds JRE, a compiler, an interpreter or loader, and several development tools in it.

4. Meaning of main method.

The main() is the starting point for JVM to start execution of a Java program. Without the main() method, JVM will not execute the program. The syntax of the main() methd is:



public: It is an access specifier. We should use a public keyword before the main() method so that JVM can identify the execution point of the program. If we use private, protected, and default before the main() method, it will not be visible to JVM.

static: You can make a method static by using the keyword static. We should call the main() method without creating an object. Static methods are the method which invokes without creating the objects, so we do not need any object to call the main() method.

void: In Java, every method has the return type. Void keyword acknowledges the compiler that main() method does not return any value.

main(): It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the main() method.

String args[]: The main() method also accepts some data from the user. It accepts a group of strings, which is called a string array. It is used to hold the command line arguments in the form of string values.

7. Java Virtual Thread.

A thread is the smallest processing unit that can be scheduled. It operates concurrently with, and mostly independently of other units of this type. It’s an instance of java.lang.Thread.

There are two kinds of threads, platform threads and virtual threads:

Platform Threads

Virtual Threads

8. Entry point method.

Java's main() method is the starting point from where the JVM starts the execution of a Java program. JVM will not execute the code, if the program is missing the main method. Hence, it is one of the most important methods of Java, and having a proper understanding of it is very important.

The Java compiler or JVM looks for the main method when it starts executing a Java program. The signature of the main method needs to be in a specific way for the JVM to recognize that method as its entry point. If we change the signature of the method, the program compiles but does not execute.

9. Meaning of System in and System out.

In Java, System.in and System.out are two commonly used members of the System class, which is part of the java.lang package. They are used for input and output operations, respectively.

System.in

Type: InputStream

Purpose: Represents the standard input stream, typically connected to the keyboard. It's used to read data from the console.

Usage: Since System.in is an InputStream, you typically need to use classes like InputStreamReader and BufferedReader to read text data more conveniently.

System.out

Type: PrintStream
Purpose: Represents the standard output stream, typically connected to the console. It's used to print data to the console.
Usage: System.out is a PrintStream, which provides convenient methods for printing data.

10. Size of data types.

Data types are divided into two groups:

- Primitive data types - includes byte, short, int, long, float, double, boolean and char
- Non-primitive data types - such as String, Arrays and Classes

Primitive Data Types

A primitive data type specifies the size and type of variable values, and it has no additional methods.

There are eight primitive data types in Java:

Data Type	**Size**	**Description**
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647

long	8bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

12. Loader in java.

Java ClassLoader

Java ClassLoader is an abstract class. It belongs to a **java.lang** package. It loads classes from different resources. Java ClassLoader is used to load the classes at run time. In other words, JVM performs the linking process at runtime. Classes are loaded into the JVM according to need. If a loaded class depends on another class, that class is loaded as well. When we request to load a class, it delegates the class to its parent. In this way, uniqueness is maintained in the runtime environment. It is essential to execute a Java program.

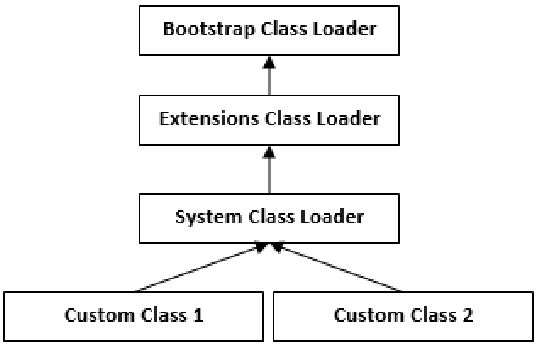


Figure: Hierarchy of Class Loader

13. Garbage collector.

    Garbage collection in Java is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.

13. Assignments