Introduction of Java

→ Java is a high-level programming language.

→ It was developed by Sun Microsystem (now owned by oracle) in the mid 1990.

→ Java is an object-oriented language that uses objects to represent data and functionality.

→ Java's main principle is "Write once, run anywhere" (WORA), meaning that java code can be written on one platform and run on any other platform that supports java.

* Some importants points of java -

- Author : James Gosling
- Vendor : Sun Microsystem (Now owned by oracle)
- Project name: Green Team
- Initial name : Oak (changed in 1995)
- Present name : Java
- Type : Open source
- Extentions : .java (Source code),
  .class (compiled code),
  .jar (archive)
- Present version: Java 18
- Operating System : Any operating system with a JVM
- Principle : WORA (Write once run anywhere)

* Where java is used?

→ Java is versatile language that can be used for wide variety of applications.

Here are some common uses of java -

1) Building web application, including server-side and client-side programming.
2) Developing mobile applications for android devices.
3) Creating desktop applications, including GUI applications and system utilities.
4) Building enterprise applications, such as customer relationship management (CRM), supply chain management system.
5) Writing software for embedded systems, such as medical devices, cars and consumer electronics.
6) Creating tools and libraries for other developers to use, such as frameworks, APIs, development kits.
7) Building scientific and technical applications.
8) Developing games and other multimedia applications.

* Parts of Java :

Java consists of three main parts -

1. J2SE / JSE (Java 2 Std Edition) → Core java
2. J2EE / JEE (Java 2 Enterprise edition) → Advance java
3. J2ME / JME (Java 2 Micro edition)

* Features of Java -

There are many features of java. Here are some of the key features of java -

- **Simple -**
  Its syntax are easy to understand and simple to learn.

- **Object - oriented -**
  Object oriented means, we organize our software as combination of different types of objects that contains both data and functionality.

- **Platform - independent and Portable -**
  It means that java code written on one platform (such as windows) can be run on any other platform (such as linux, mac) without modification.

- **Architecture - Neutral -**
  It means that, it can run on a variety of hardware architecture without requiring modification to the code.

- **Robust -**
  Java is designed to be robust (strong) and error - free.

- Secure -

  It is secure, as it cannot get affected by virus or any malware because of JVM as it has a secures/ verifier.

- Multithreaded -

  Java include support for multithreaded programming, which allows multiple thread to run concurrently within the same program.

- Distributed -

  It supports distributed computing which is mostly used by business globally where multiple business are connected to a single network.

- Dynamic -

  It supports dynamic linking and loading, which allows code to be loaded at runtime rather than at compile time. This enables a wide range of dynamic programming Techniques.

* First program using Notepad and compilation and execution through CMD -

```
import java.lang.*;
Public class MyFirst
 {
      Public static void main(String args[])
       {
          System.out.println("Hello World");
       }
 }
```

→ Steps for execution and compilation -

i) Save the file (Write file name in double quote)
   Eg:- "MyFirst.java"

Note - While saving the file, use same name as class name.(MyFirst)

ii) Go to command prompt -

→ Write `javac MyFirst.java` for compilation process.
(If some errors are occure then it would show)

→ After compilation, if we chick the directory then there are two files ~~are~~ ≠ will be there -

1) MyFirst.class (This file generated by compiler)
2) MyFirst.java (This is that file which we have written.)

iii) Write `java MyFirst` for running the program.

→ It will print Hello world

* Exploration of skeleton of java program -

- import java.lang.*;
  It imports the java pacages file.
- Class is a keyword is used to declare a class in java.
- Public keyword is an access specifier which represents visibility, it means it is visible to all.
- Static is a keyword, if we declare any method as static, it is known as static method. The main advantage of static method is that there is no need to create object to invoke the static method.
- Void is the return type of the method, it means it doesn't return any value.
- main is a method, it represents startup of the program. The main method is executed by the JVM
- String args[] is used to command line argument.

String - 'S' is capital

→ It can be written in 3 ways -

1) String args[]
2) String [] args
3) String []args

- System.out.println() is used to print statement (Hello world).

  - System: It is a class, which belongs to java.lang package.

  - out: It is an output stream object, which is a member of System class.

  - println(): It is a method to display any kind of output on the screen.

* Comments in Java -

The comments are statement that are not executed by the compiler and interpreter.

→ Comments are used to describe the purpose of the code, make the code easier to understand and leave notes for future programmers who may need to modify or maintain the code.

* Types of comment -

1. Single-Line Comment - This is used to comment only one line.

Syntax :- // This is single line comment.

ii) Multi-line Comment -

Syntax :-
```
/* This
is
multiline
comment */
```

* Process behind the execution of Java file -

- Writing the java code -

  Whatever program we write, it is called as "Source Code" which always have extention ".java".

- Compiling the code -

  This source code is translated or compiled into byte code (.class) using a java compiler. Although this bytecode is not executable but it is a error free program.

- Running the code (Main or Actual step) -

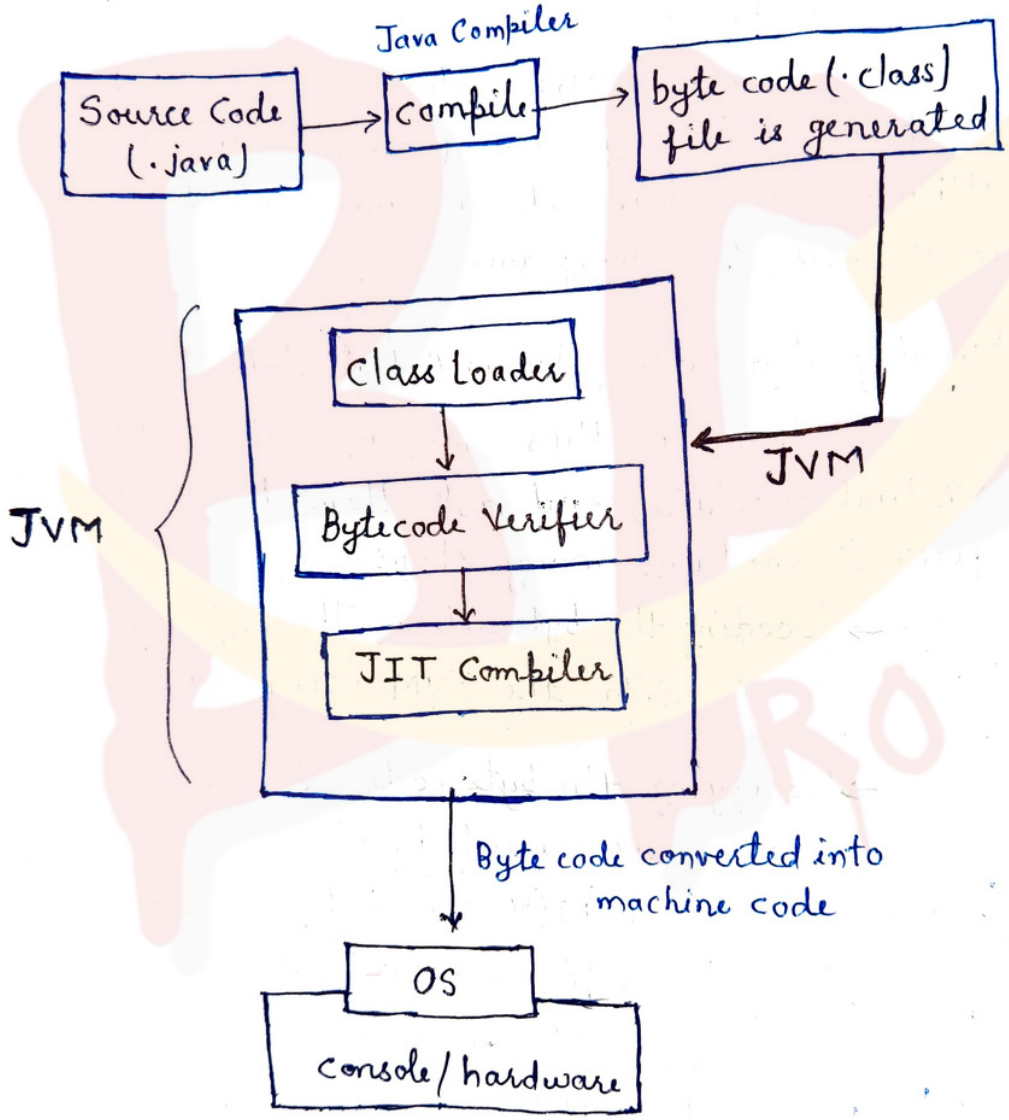
  To run this code, we call a JVM (Java Virtual machine) and ask them to execute this file. It contains many steps -

  → Loading the byte code - The byte code is loaded into the JVM using class loader.

  → Verifying the byte code - Before the bytecode is executed, it is verified by the JVM (verifier) to ensure that it is safe and does not violate any security rules.

  → Executing the byte code - This JVM will have an interpreter or JIT (Just in time) compiler that converts this byte code into machine code line by line and get it executed on the hardware (console).

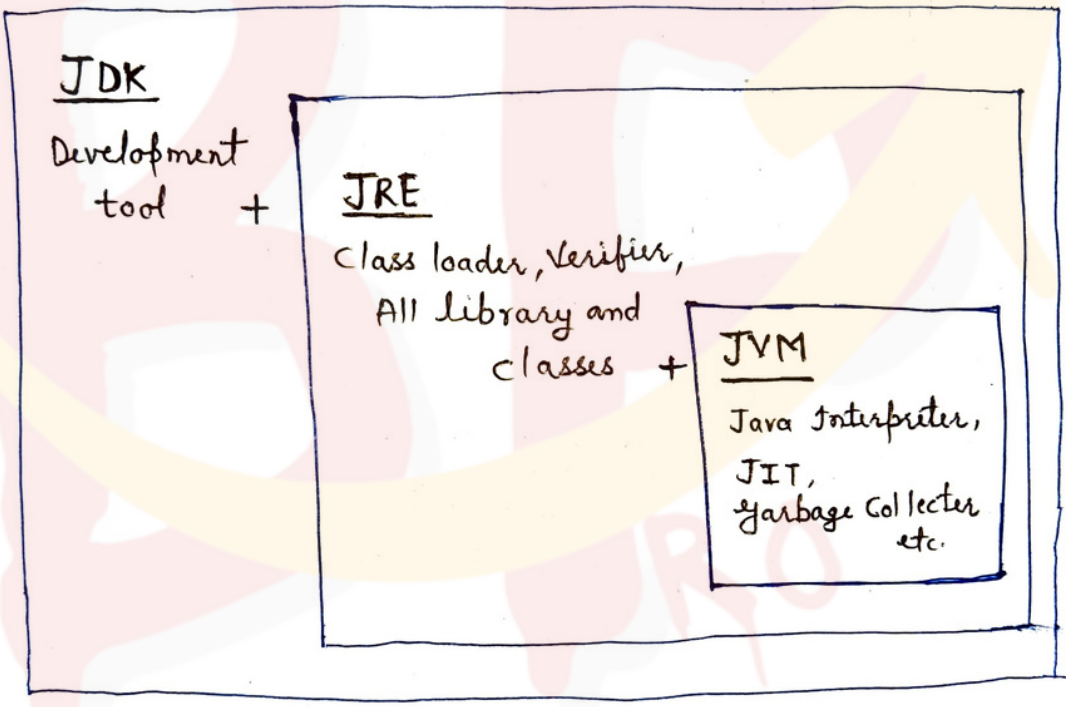
Notes: JVM can not directly interact with the console, it needs to interact via a software (operating system).



* Java Development Kit (JDK) :

If we want to develop and execute java program in our system, we have to install JDK (JRE, JVM, supporting tools).

→ It is used for developing java programs, so it will have all development tools that are useful for compiling or debugging java programs and various other tools are available.



* JDK contains JRE and JRE contains JVM, so when we install JDK then we also install or get JRE and JVM along with it.

* JRE (Java Runtime Environment) -

Once we have compiled a program, we have to execute it, so for executing the java program, JRE is used.

→ The process of execution is done by actually JVM which is the part of JRE.

→ So we say that our java program are running inside JRE that are actually being executed by JVM.

Interview Questions

Q.1) What is Bytecode ?

When a java program is compiled, the java compiler translates the java source code into Bytecode, which is highly optimized set of instruction. It is an intermediate representation of the program that is designed to be executed by a JVM on any platform that has JVM installed.

2) How java is platform independent ?

Java programs are compiled into bytecode, which is highly optimized set of instructions that can be executed on any platform that has JVM installed. This means that the compiled bytecode is plateform independent and can be executed on any platform that supports the JVM.

3) Can we change the syntax of main method ?

No, we cannot change the syntax of the main method in java. The syntax of the main method is fixed and must be followed exactly as specified in the java language specification.

```
public static void main (String[] args)
{
    // code
}
```

Since JVM is responsible to execute every java program and JVM knows only main method. If main method is not there or its syntax is changed, our program will not executed.

4) Can we make main method as private?

No, we cannot keep the main method as private because JVM requires access to the main method from outside the class. Since private fields can not be accessed from outside the class, if we keep the main method as private, the program will not be executed and it will generate an error message stating that the main method is not found in the given class.

5) Is java enables high performance and how?

Yes, java enables the high performance as java has Just-In-Time (JIT) compiler, which can compile same functionality bytecode in same time. A java programs takes less time to compile than other programming language's compiler.

6) What is Garbage Collection?

Garbage collection is an automatic memory management feature that manages the allocation and deallocation of memory in java. It is responsible for reclaiming memory that is no longer in use by the program and making it available for future use. It runs automatically in the background and checks for objects that are no longer being used by program.