

INTRODUCTION TO JAVA

- **Java** is a widely used **high-level, robust, object-oriented, secure programming language and platform**. Java was created in **1995** by **Sun Microsystems** (now an Oracle company). **James Gosling** is regarded as the “**Father of Java.**” It was called Oak before Java by Oracle. The name Oak was changed to Java by Oracle’s James Gosling and his team because Oak was already a registered company. In this article, we will cover the Introduction to java.
- Java is an object oriented, case sensitive, class based programming language that is portable and platform-independent, meaning that Java code can run on any device with a **Java Virtual Machine (JVM)** installed.

❖ Features In Java Programming

Let’s see the various features provided by Java.

1. Object Oriented Programming (OOPs)

Object-oriented programming (OOPs) organizes software design and logic around objects, classes, and data rather than procedures or functions. Object-oriented programming means to leverages real-world techniques to develop application code while keeping data security and integrity at the forefront of design. There are **four fundamental** pillars of Object-oriented programming, and these are as follows:

- **Encapsulation:-** It is a term used in Object-Oriented Programming to denote the practice of combining data and its associated functions into a single unit. This is similar to a real-world scenario in a firm, where separate departments such as accounts, finance, and sales conduct specific responsibilities and retain relevant data records.
- **Abstraction:-** Abstraction means showing only the most important information while obscuring the details. Data abstraction is the process of presenting only essential information about data to the outer world while obscuring the background details or implementation.
- **Polymorphism:-** Polymorphism is defined as having several forms. Polymorphism is defined as the ability of data to be displayed in more than one form.
- **Inheritance:-** In Object-Oriented Programming, Inheritance refers to the ability of a class to acquire properties and attributes from another class. It is a fundamental feature that plays a crucial role in designing and implementing object-oriented systems.

2. Platform Independent in Java Programming

In Java programming , the compiler compiles source code into bytecode, which is then executed by the Java Virtual Machine (JVM). A wide range of systems, including Windows, Linux, and macOS, can run this bytecode generated by the compiler because each platform has its own JVM, which generates platform-specific machine code from the same bytecode. As a result, running the same source code on different platforms with different JVMs produces similar results, making Java a platform-independent language.

3. Simple

Java is developed in such a way that it is simple to understand. It is simple to master Java if you understand the fundamentals of OOPs. Furthermore, it lacks complex features such as operator overloading, pointers, multiple inheritances, etc.

4. Java Programming is Safe and Reliable

The fact that we don't have access to pointers in Java makes it incredibly safe. It is impossible for problems like buffer overflow or stack corruption to arise because we cannot access external arrays due to the lack of pointers.

5. Robust

The Java programming language is robust, which implies it is reliable. It is designed in such a way that it makes every attempt to find errors as early as possible, which is why the Java compiler can detect errors that other programming languages cannot. Garbage collection, exception handling, and memory allocation are the core aspects of Java that make it robust.

6. High Performance

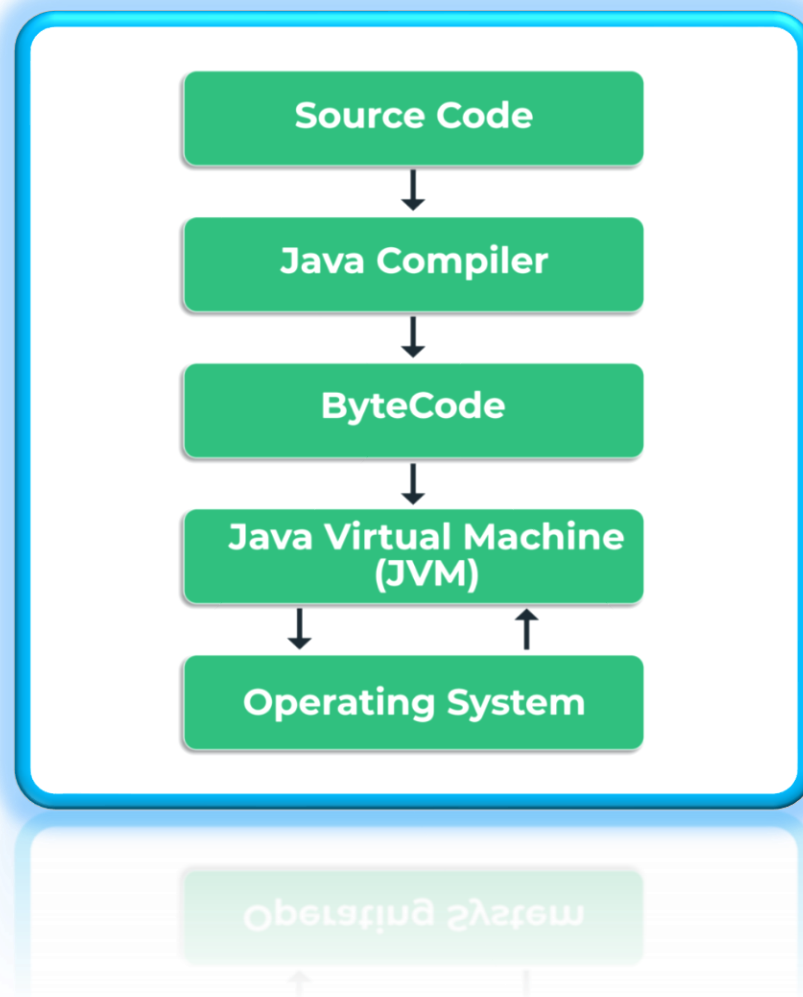
The architecture of Java is designed to minimize runtime overhead. To achieve this, Java sometimes utilizes a Just-In-Time (JIT) compiler, which compiles code on-demand basis. The JIT compiler only compiles those methods that are called during runtime, which results in faster execution of the application.

7. Sandbox Execution

Java applications operate in a distinct environment that enables users to run their programs without impacting the underlying system. This is accomplished by using a bytecode verifier, which ensures that the code is free of errors and enhances security by checking for any unauthorized access.

❖ Java Programming Terminology

Let's try to understand the various terms frequently used in Java and how the code in Java is executed. The flow chart given below shows how a Java program is executed.



1. Java Virtual Machine (JVM)

JVM is the common name for this. Program execution is divided into three stages. The program is written, compiled, and executed.

- A Java programmer, such as you and I, write programs.
- The JAVAC compiler, a primary Java compiler included in the Java development kit (JDK), performs the compilation. It produces bytecode as output and accepts a Java program as input.
- A program's JVM runs the bytecode produced by the compiler during the Running phase.

2. Bytecode in The Development Process

Bytecode is a platform-independent set of instructions that the JVM understands and executes. When we compile any Java code, the compiler converts it into bytecode. It is equivalent to the assembler in C++. Bytecode is an intermediary level of instructions between the high-level language in which we write code and the low-level machine code that the computer executes. When we write code in a high-level language, it is compiled into bytecode. The JVM then translates the bytecode into a low-level set of instructions.

3. Java Development Kit (JDK)

As the name implies, Java Development Kit is a full-time kit that includes a compiler, Java Runtime Environment (JRE), Debuggers, and Java documentation. We must have JDK installed on our computers in order to create, compile, and run the Java program.



4. Java Runtime Environment (JRE)

The Java Development Kit (JDK) includes the Java Runtime Environment (JRE). Installing JRE on a computer enables the execution of Java programs, but does not provide the ability to compile them. JRE includes JVM, a browser, applet support, and plugins that are necessary for running Java applications. As a result, having JRE installed on a computer is required in order to run Java programs.

What is Java Garbage Collector in Java Programming?

- In Java, the Java Virtual Machine (JVM) features a Garbage Collector that automatically frees up memory by recollecting items that are no longer referenced in the code.
- This makes life easier for programmers because they no longer have to manage memory manually.

- However, programmers must still be careful about what they write in their code and how long they use certain objects because the Garbage Collector cannot recover memory for objects still in use
- Garbage collection is a hands-free process that happens automatically.

➔ First Program In Java

To run Java Code on your device, you must install the **Java Development Kit** (JDK) on your Windows computer.

Let's see the program in Java for printing **"Welcome to Topperworld"**.

Program:

```
// Importing important classes from packages
import java.io.*;

// Main class
public class Topperworld {
    public static void main(String[] args)
    {
        // Print statement
        System.out.println("Welcome to Topperworld");
    }
}
```

Output: **Welcome to Topperworld**

Explanation:

1. Comments:- Comments are used to explain code and are used similarly in Java, C, and C++. Compilers ignore comment entries and do not execute them. Comments might be on a single line or multiple lines.

Syntax

- *For Single Line*

```
// This is comment
```

- *For Multiple Line*

```
/* This is comment */
```

2. import java.io:- The “**import**” keyword in Java includes a class in the code. When we import classes from the “io” package, we get access to input and output streams that allow us to read or write files and input or output.

3. Class:- In Java, a class is a container that holds data and functions for a program. The behaviour of the class is defined by its functions. The class named **Topperworld** has only one function, “**Main**”.

4. Public Static:- Static methods in Java can be called without creating objects. Instead, they can be executed using the dot operator with the name of the class.

5. Void:- This keyword denotes that the method will not return anything.

6. main():- It is the entry point of our program.

7. System.out.println:- It is used to print the content.

❖ Advantages of Java Programming

- **Platform Independent:-** A wide range of systems, including Windows, Linux, and macOS, can run the bytecode generated by the compiler because each platform has its own JVM, which generates platform-specific machine code from the same bytecode. This makes Java a platform-independent language.
- **Simple:-** Java is developed in such a way that it is simple to understand. It is simple to master Java if you understand the fundamentals of OOPs.
- **Object-oriented Programming:-** Java is an object oriented programming language, which implies it follows encapsulation, inheritance, and polymorphism concepts.
- **Secure:-** Java contains built-in security features that make it a secure environment for developing projects, such as automatic memory management and type checking.
- **Significant developer community:-** Java has a huge and active developer community, which means a lot of help is available for learning and utilizing the language, making it a good language for development.