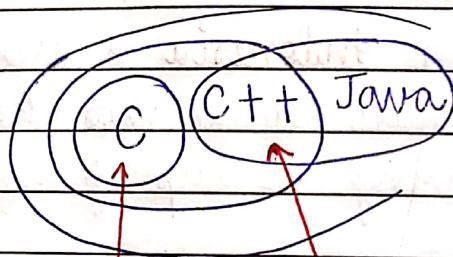


Core and Advance JAVA

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems in 1991. It took 18 months to develop the first working version. This language was initially called 'Oak' but was renamed Java in 1995.

The language derives much of its syntax from C and C++ but is simpler to use than C++.

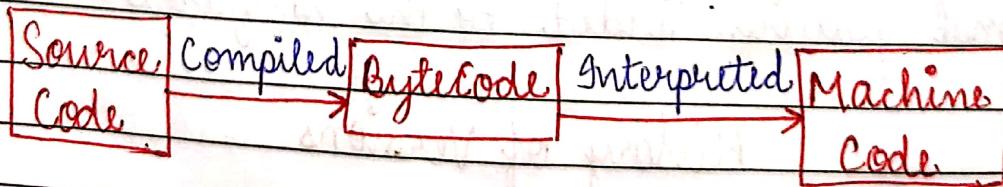
Java is a pure Object Oriented Language.



Procedure Object-Oriented
Oriented

How Java Works?

Java is compiled into the byte code and then it is interpreted to machine code.



History of Java

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of Sun engineers called Green Team.



Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.

Firstly, it was called Greentalk by James Gosling.

After that it was called Oak and was developed as a part of the Green Project.

In 1995, Oak was renamed as Java because it was already a trademark by Oak Technologies.

Java is an island in Indonesia where the first coffee was produced. Java name was chosen by James Gosling while having a cup of coffee nearby his office.

Initially developed by James Gosling at Sun Microsystems and released in 1995!

JDK 1.0 was released on January 23, 1996. After the first release of Java, there have been many additional features added to the language.

History of Versions

JDK 1.0	January 1996
JDK 1.1	February 1997
JDK 1.2	December 1998
J2SE 1.3	May 2000
J2SE 1.4	February 2002



Java SE 5.0	September 2004
Java SE 6	December 2006
Java SE 7	July 2011
Java SE 8	March 2014
Java SE 9	March September 21, 2017
Java SE 10	March 20, 2018
Java SE 11	September 25, 2018
Java SE 12	March 19, 2019
Java SE 13	September 17, 2019
Java SE 14	March 17, 2020
Java SE 15	September 15, 2020
Java SE 16	March 16, 2021
Java SE 17	September 14, 2021
Java SE 18	March 22, 2022
Java SE 19	September 20, 2022
Java SE 20	March 21, 2023
Java SE 21	September 19, 2023

Features of JAVA

The features of JAVA are →

Simple

Secure

Portable

Object - Oriented

Robust

Multithreaded

Architecture - Neutral

Interpreted

High Performance

Distributed Dynamic

Simple → Java was designed to be easy for the professional programmer to learn and use effectively.

If you already understand the basic concepts of object-oriented programming, learning Java will be even easier.

Object-Oriented → Although influenced by its predecessors, Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean, usable, pragmatic approach to objects.

Secure → When it comes to security, Java is always the first choice.

With java secure features it enable us to develop virus free, temper free system.

Java program always runs in Java Environment with almost null interaction with System's Operating System, hence it is more secure.

Portable → Java takes minimum memory because of bytecode.

It can be moved from one machine to another very effectively.



Robust → Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking.

But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing **Garbage Collector** and **Exception Handling**.

Multithreaded → Java Multithreading feature makes it possible to write program that can do many tasks simultaneously.

Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time.

Architecture - Neutral → Compiler generates bytecodes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

Interpreted → Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode. This code can be executed on any system that implements the Java Virtual Machine.

High Performance → Java uses the concept of JIT (Just-in-time) compiler. It reduces the waiting time of converting bytecode into machine code. Due to which the overall execution of Java program becomes faster.

Distributed → Java is designed for the distributed environment of the Internet because it handles TCP/IP protocols. In fact, accessing a resource using a URL is not much different from accessing a file.

Dynamic → Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time. This makes it possible to dynamically link code in a safe and expedient manner.

Platform Independent → Java program can run on any OS. Java compiler (javac) compiles source code file (.java) into bytecode (.class) & this bytecode can run on any platform, making Java as an platform independent language.

Difference between C++ and JAVA

C++

It is Platform Dependent. It is Platform Independent

It supports goto statement, Operator Overloading, Pointer and Multiple Inheritance.

JAVA

It does not support goto statement, Operator Overloading, Pointer, Multiple Inheritance.



It uses a compiler.

It uses compiler and interpreter.

It supports call by value and call by reference.

It supports only call by value.

Memory Management is done using a pointer.

Memory Management is done by the system.

Runtime Error Detection mechanism is the responsibility of the programmer.

Runtime Error Detection is the responsibility of the system.

Libraries have low-level functions.

Libraries have high-level functions.

It allows procedural programming as well as object-oriented programming.

It allows only Object-Oriented Programming Model.

It has a scope resolution operator (::), structure, union.

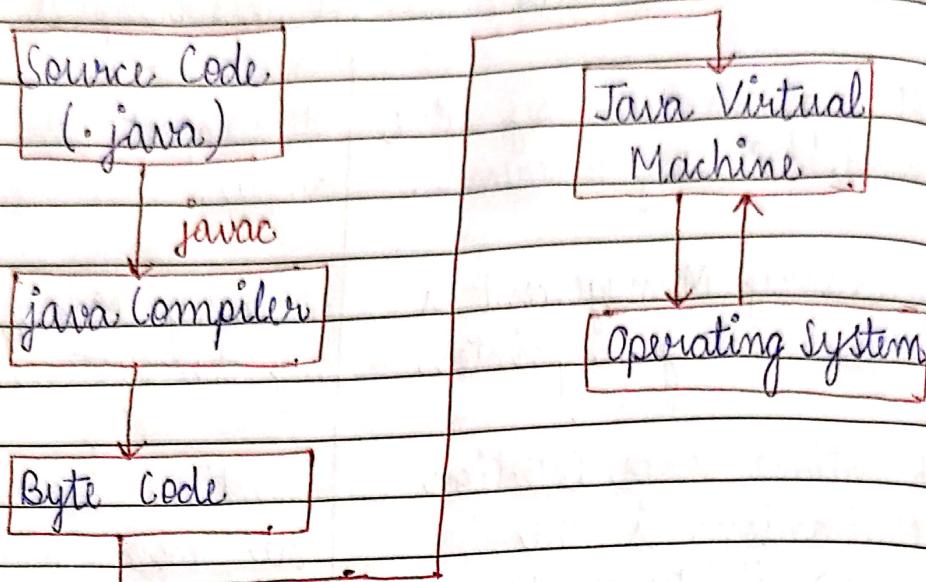
It does not have a scope resolution operator (::), structure, union.

It has new and delete keyword for object management.

It has automatic Garbage Collection for object management.



JAVA Architecture



Compilation and Interpretation in JAVA → It combines both the

approaches of compilation and interpretation.

First java compiler compiles source code into bytecode at the runtime. Then, JVM interprets this bytecode & generates machine code which will be directly executed by the operating system.

Components of Java Architecture → There are three main components

of Java Language → JVM, JRE and JDK.

a) JVM (Java Virtual Machine) → Java applications are called **WORA**

(Write Once, Run Anywhere) because of their ability to run a code on any platform. This is done by JVM.

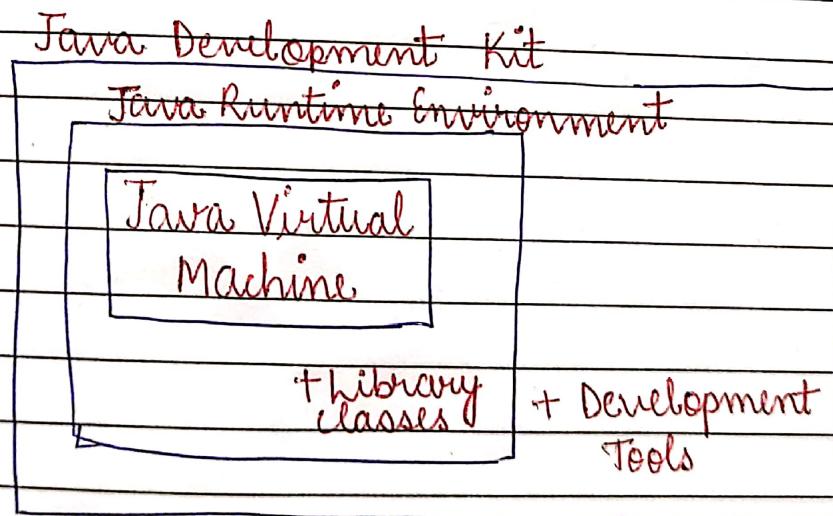
The JVM is a Java Platform component that provides an environment for executing Java programs.



JVM interprets the bytecode into machine code which is executed in the machine in which the Java program runs.

(b) **Java Runtime Environment** → The JRE is the on-disk system that takes your Java code, combines it with the needed libraries, and starts the JVM to execute it. The JRE contains libraries and software needed by your Java programs to run.

(c) **Java Development Kit** → The JDK is a software development environment used to develop Java applications and applets. It contains JRE and several development tools, an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) accompanied with another tool.



$$\text{JDK} = \text{TRE} + \text{Development Tool}$$

$$\text{TRE} = \text{JVM} + \text{Library classes}$$