## Introduction to Java

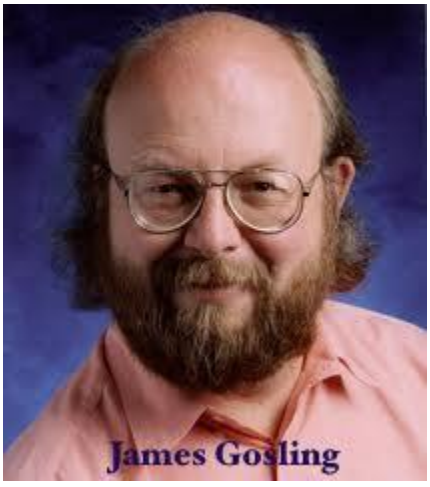
- Java is a computer programming language. It enables programmers to write computer instructions using English-based commands instead of writing in numeric codes.
- It's known as a "**high-level**" language because humans can read and write easily. Like English, Java has a set of rules that determine how the instructions are written. These rules are known as its "syntax". Once a program has been written, the high-level instructions are translated into numeric codes that computers can understand and execute.
- Java is an **object-oriented** programming language.

## Java History

**The history of Java** is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of java starts with Green Team. Java team members (also known as **Green Team**), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was suited for internet programming. Later, Java technology was incorporated by Netscape.

The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted, and Dynamic".

Currently, Java is used in internet programming, mobile devices, games, e-business solutions, etc. Significant points describe the history of Java.



- **James Gosling**, **Mike Sheridan**, and **Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers is called the **Green Team**.
- Originally designed for small, embedded systems in electronic appliances like set-top boxes.
- Firstly, it was called **"Green Talk"** by James Gosling, and the file extension was .gt. After that, it was called **Oak** and was developed as a part of the Green project.
- **Why Oak?** Oak is a symbol of strength and has been chosen as a national tree in many countries like the U.S.A., France, Germany, Romania, etc.
- In 1995, Oak was renamed **"Java"** because it was already a trademark by Oak Technologies.
- Java is an island of Indonesia where the first coffee was produced (java coffee).
- Notice that Java is just a name, not an acronym.
- Initially developed by James Gosling at Sun Microsystems (now a subsidiary of Oracle Corporation) and released in 1995.
- In 1995, Time magazine called **Java one of the Ten Best Products of 1995**.
- JDK 1.0 was released on (23 January 1996).

## Java Version History

Many Java versions have been released. The current stable release of Java is Java SE 10.

JDK Alpha and Beta (1995)

JDK 1.0 (23rd Jan 1996)

JDK 1.1 (19th Feb 1997)

J2SE 1.2 (8th Dec 1998)

J2SE 1.3 (8th May 2000)

J2SE 1.4 (6th Feb 2002)

J2SE 5.0 (30th Sep 2004)

Java SE 6 (11th Dec 2006)

Java SE 7 (28th July 2011)

Java SE 8-**LTS-Long Term Support** (18th Mar 2014)

Java SE 9 (21st Sep 2017)

Java SE 10 (20th Mar 2018)

**Java SE 11-LTS (September 2018)**

Java SE 12 (March 2019)

Java SE 13 (September 2019)

Java SE 14 (Mar 2020)

Java SE 15 (September 2020)

Java SE 16 (Mar 2021)

**Java SE 17-LTS (September 2021)**

Java SE 18 (March 2022)

Java SE 19 (September 2022)

Java SE 20 (March 2023)

**Java SE 21-LTS (September 2023)**

Java SE 22 (March 2024)

Java SE 23 (September 24)

Java SE 24 (March 24)

**Java SE 25-LTS (September 25)**

Since the Java SE 8 release, the Oracle corporation has followed a pattern in which every even version is released in March, and an odd version is released in September.

**Where is it used?**

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as Acrobat Reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in etc.
3. Enterprise Applications such as banking applications.
4. Mobile Application
5. Smart Card, Debit, Credit Cards

6. Robotics
7. Games
8. Television set-top boxes, Coffee Machine, Toaster machine, etc.

**Types of Java Applications**

There are mainly four types of applications that can be created using Java programming:

**1) Desktop Application**

It is also known as a desktop application or window-based application. We must install an application on every machine, such as a media player, antivirus, etc. AWT and Swing are used in Java to create standalone applications.

**2) Web Application**

An application that runs on the server side and creates dynamic pages is called a web application. Currently, servlet, jsp, struts, JSF, etc. technologies are used to create web applications in Java.

**3) Enterprise Application**

An application that is distributed in nature, such as banking applications, etc. It has the advantage of high-level security, load balancing, and clustering. In Java, EJB is used to create enterprise applications.

**4) Mobile Application**

An application that is created for mobile devices. Currently, Android and Java ME are used to create mobile applications.

**Java Editions/Platforms**

There are four platforms or editions of Java:

**Java Standard Edition (JSE):** The Java Standard Edition (Java SE) is for building desktop applications and applets. These applications typically serve only a small number of users at one time. It is a Java programming platform. It includes Java programming APIs such as Java. lang, java.io, java.net, and java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

**Java Enterprise Edition (JEE):** It is an enterprise platform, which is mainly used to develop web and enterprise applications. It is built on the top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

**Java Micro Edition (JME):** The Java Micro Edition is for mobile applications (e.g., cell phones, PDAs) and embedded devices (e.g., TV tuner boxes, printers).

**JAVA Card:** used in SIM Cards, ATM Cards, Credit Cards, and SMART Card reader programming.

## JAVA FEATURES

Java was designed with a few key principles in mind:

1. **Simple:**
   - It resembles with C++.
   - No need for header files like conio. h, iostream.h, etc.
   - No pointers, structures, union, operator overloading, virtual base class, friend function, etc.
   - The size of the basic interpreter and class support is about 40k bytes.
   - The basic standard libraries and thread support add 175k bytes only.
   - Java provides a bug-free system due to its strong memory management.
   - It also has an automatic memory allocation and deallocation system.
2. **Object Oriented:**

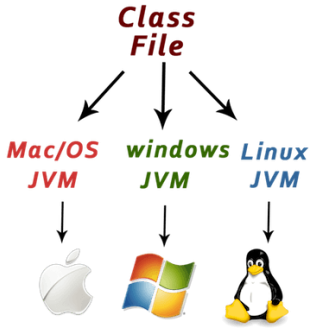
   It supports all the features of Oops. Some of the important object-oriented features are namely:

- Class
- Object
- Inheritance
- Interfaces
- Packages
- Data Abstraction
- Data Encapsulation
- Polymorphism

- Overloading
- Reusability

**3. Platform Independent and Architecture Neutral:**

- The concept of write-once-run-anywhere (WORA- known as the Platform Independent) is one of the key features of Java that makes Java the most powerful language.
- The programs written on one platform can run on any platform, provided the platform must have the JVM.




- Java was written to be a portable language that does not care about the operating system or computer hardware.
- The size of primitive data types is independent.
- Architecture-neutral
- Java is architecture-neutral because there are no implementation-dependent features; for example, the size of primitive types is fixed.
- In C programming, the int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. For example, Borland C++ takes 4 bytes, and Turbo C++ takes 2. However, JAVA occupies 4 bytes of memory for 32- and 64-bit architectures in Java.

**4. Distributed:**

- Java supports many networking protocols like TCP, IP, HTTP, and FTP, and Java has an extensive library of these protocols.
- Java applications can open and access objects across the internet via urls. Internet programmers can call functions on these protocols and access the files from any remote machine on the internet rather than writing codes on their local system.

- Java Support networking capabilities like socket connection, common gateway Interface (CGI) scripting, Servlet, JSP, and JSF- make server-side processing in Java efficient.
- Java is distributed because it facilitates users in creating distributed applications in Java. RMI and EJB are used to create distributed applications. This feature of Java allows us to access files by calling the methods from any machine on the internet.

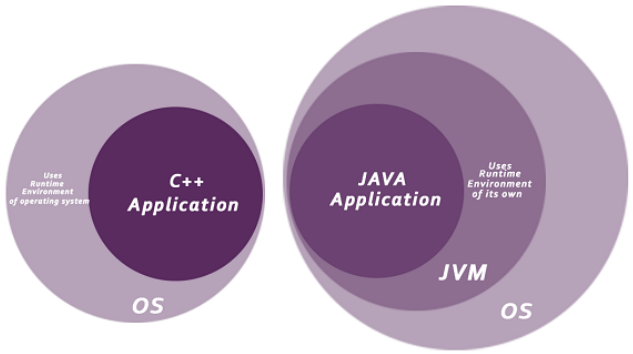
5. **Reliable and Robust:**

- Java has a strong memory allocation and automatic garbage collection mechanism that solves all the memory management problems.
- It provides powerful exception handling, which captures errors and eliminates any risk of crashing.
- Provide compile-time and run-time checking -The compiler checks the program to see whether there are any errors, and the interpreter checks any run-time errors and makes the system secure from the crash.

6. **Secure:**

- Java does not support a pointer like C/C++. Hence, the memory locations of the system cannot be accessed through a JAVA program, so any program developed in Java cannot be used to hack a system.
- Java uses the public key encryption system to allow the Java applications to transmit over the internet in a secure encrypted form.
- The byte code Verifier checks the classes after loading.

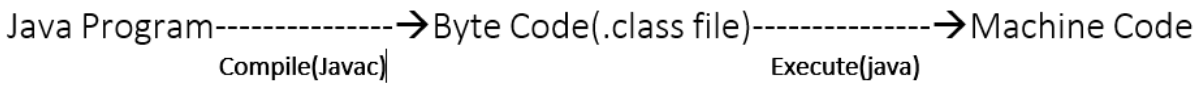
  Java programs run inside the virtual machine, so Java is probably the most secure programming language to date.



7. **Portable:**

- The feature Write-once-run-anywhere (WORA) means the Java compiler generates byte code instructions that can be implemented on any machine. Java is portable because it allows you to carry the Java bytecode to any platform. It doesn't require any implementation.
- The size of JAVAC and JAVA is approximately less than 100KB.


8. **Compiled and Interpreted:**




9. **High-performance**

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language, so it is slower than compiled languages, e.g., C, C++, etc.

10. **Popular:**

- Java has rich libraries like applets, servlets, etc that make it popular.
- In C++ Library→ Customize→Application
- In Java Library→ Application

11. **Multithreaded**:

- Java language provides an environment by which several tasks can be initiated and managed easily such a feature is called multithreading. Multithreading means a single program having different threads executing independently at the same time.

- Multiple threads execute instructions according to the program code in a process or a program.
- The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area.
- Example: In the MS Word program, create a different thread for functions like Save, Scroll, Line Spacing, font size, auto-save, spell check, delete, object insert, etc.
- Multithreading works similarly to multiple processes run on one computer.

12. **Dynamic and Extensible:**

- Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand.
- Java can dynamically link in new classes, libraries, methods, and objects. Java programs support functions written in another language, such as c or C++. These functions are known as native methods; this facility enables the programmers to use the efficient functions available in this language; native methods are linked dynamically at runtime.

The Sun Microsystems team successfully combined these key principles, and Java's popularity can be traced to its robust, secure, easy-to-use, and portable language.

**Java Virtual Machine (JVM):**

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist.

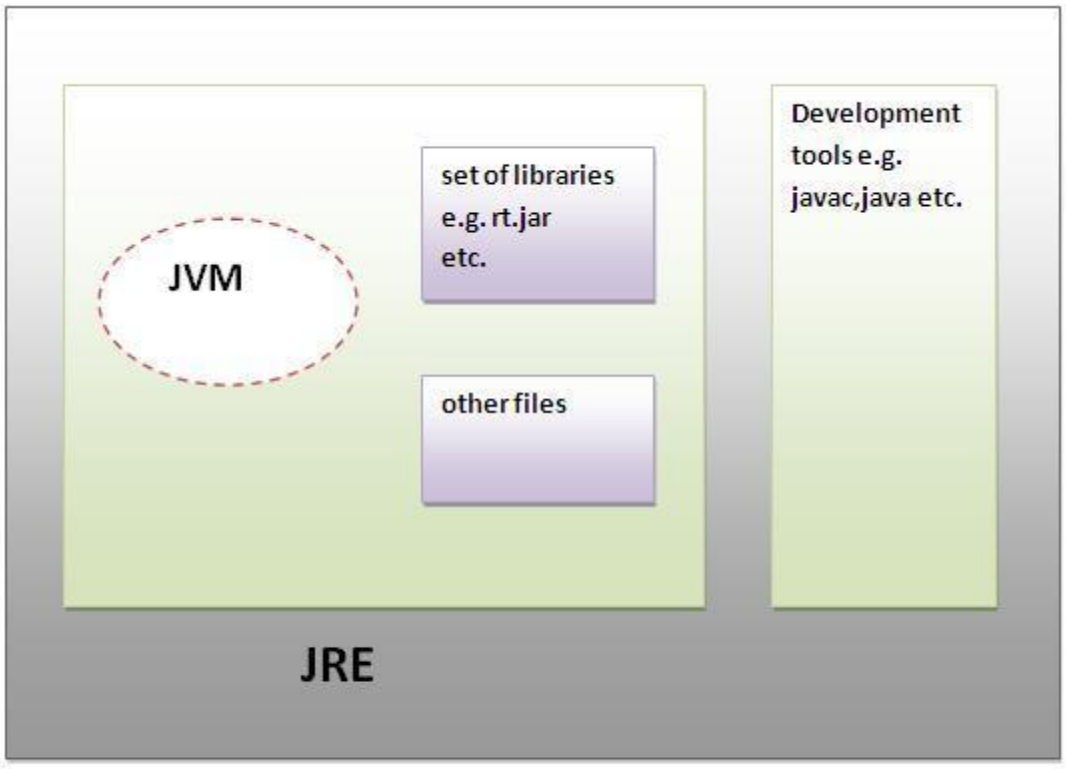
The specification provides a runtime environment in which Java bytecode can be executed.

JVMs are available for many hardware and software platforms.

JVM, JRE, and JDK are platform-dependent because the configuration of each OS differs. But Java is platform-independent.

The JVM performs the following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment




**Java Runtime Environment (JRE):**

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a software tool for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

JDK=JRE + Development Tools (JAVAC, JAVA)

**Java Development Kit (JDK):**

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment for developing Java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform

The JDK contains a private Java Virtual Machine (JVM) and a few other resources, such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc., to complete the development of a Java Application.

**Example:**

JDK 1.5,JDK 1.6 ,JDk 1.8…….23.

**Note:**

The JRE is contained within the JDK (i.e., if you download the JDK, you will be able to create and run Java programs.).