What is java?

Java is a renowned and most-used programming language worldwide. It is a general-purpose, class-based, distributed, and object-oriented programming language. It is designed to have as lesser implementation dependencies as possible.

**Characteristics of Java**

The following are the characteristics of Java:

1. Java is an object-oriented programming language.
2. It has automatic garbage collection.
3. Java generates a bytecode using source code which makes it platform-independent.
4. Pointers don’t exist in Java. However, references do.
5. Java does not support multiple inheritances using classes. However, it can be achieved using interfaces.
6. Operator overloading is not allowed in Java.
7. Java doesn’t support structures and unions like C++.

**Features of Java**

Following are a few of the Features of Java Programming:

1. **Object-Oriented:**Java is an object-oriented programming language, which means it is based on the concept of objects and classes. Objects contain data and methods that operate on that data, while classes define the objects and their behavior.
2. **Platform-Independent:**Because Java bytecode is platform-agnostic, you can compile a Java program on one operating system and run it on another. For example, you could write a Java program on a Windows machine and run it on a Linux machine without any changes.
3. **Secure:**Java has a number of security features built into the language to help protect against common vulnerabilities. For example, you can use the "final" keyword to make a variable read-only, which helps prevent accidental modification.
4. **Multithreaded:**In Java, you can create multiple threads of execution within a single program to allow for concurrent processing. For example, you could create a thread to handle incoming network connections and another thread to perform data processing.
5. **High-Level:** Java is a high-level programming language, which means it is easier to read and write compared to low-level languages such as assembly. It also has a large standard library that provides many useful functions and data structures.
6. **Robust:**Java strongly emphasizes exception handling and type safety, which helps prevent errors and make the code more robust. For example, you can use try/catch blocks to handle exceptions and prevent the program from crashing. Java also has a garbage collector that automatically frees up memory that is no longer being used, which helps prevent memory leaks.

JVM, JRE, JDK

The Java Virtual Machine (JVM) is a software component that runs on a computer and allows Java programs to be executed. The JVM is responsible for interpreting Java bytecode, which is a compiled form of Java code that is produced when a Java program is built.

Here's how the JVM works:

1. When a Java program is executed, the JVM loads the bytecode for the program into memory.
2. The JVM then interprets the bytecode, translating it into instructions that the computer's processor can understand and execute.
3. As the JVM interprets the bytecode, it also performs various tasks such as allocating memory for the program, managing threads (which allow the program to perform multiple tasks concurrently), and handling exceptions (which occur when something goes wrong during execution).
4. The JVM is also responsible for garbage collection, which is the process of reclaiming memory that is no longer being used by the program.

In short, the JVM plays a crucial role in the execution of Java programs, providing a runtime environment that allows Java code to be executed on a wide range of devices and platforms.

**How is Java a Platform-Independent Language?**

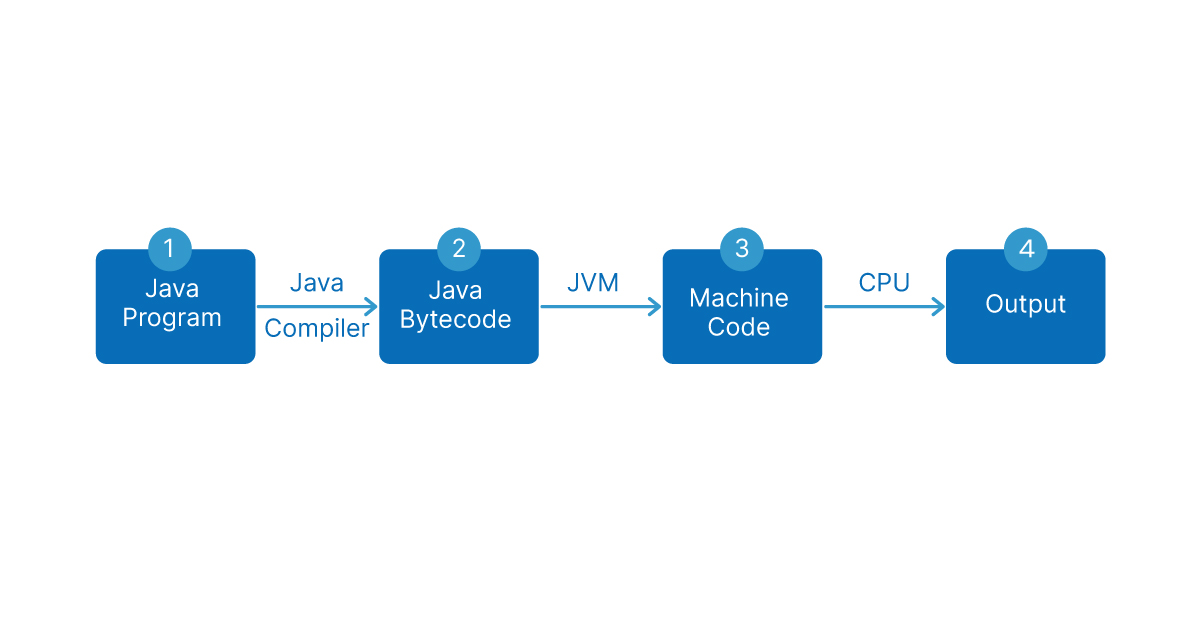
Java is a platform-independent language because it can run on any device or operating system that has a Java Virtual Machine (JVM) installed. When a Java program is compiled, it is converted into a form called bytecode. This bytecode is not specific to any particular operating system or processor architecture, and can be run on any device that has a JVM installed.

The JVM is responsible for interpreting the bytecode and executing it on the specific device or operating system. This means that the same Java program can be run on any device or operating system that has a JVM installed, without the need to recompile the code for each specific platform.

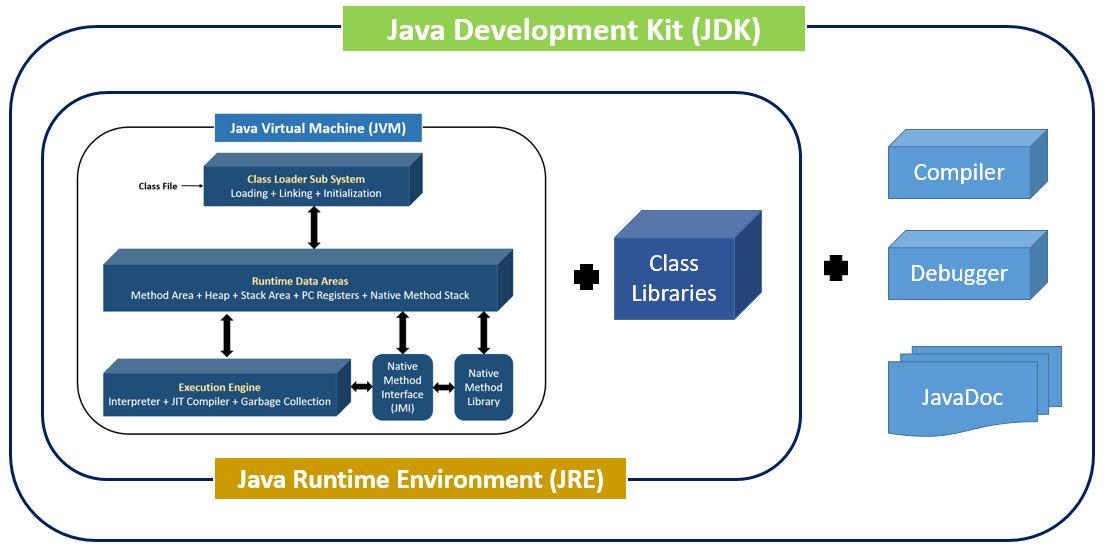
For example, a Java program that was compiled on a Windows computer can be run on a Mac, Linux, or Android device without any changes, as long as the device has a JVM installed.

Java Architecture can simply be explained in the following steps:

1. First, the source code is converted into bytecode using Java compiler.
2. Then, the bytecode is gets converted into machine code by Java Virtual Machine (JVM).
3. Afterward, the machine code goes to CPU for processing and generates an output.
4. The output will be displayed on the screen.

**JDK = Development Tools + (JRE + JVM)**

**JDK = Development Tools + (JRE + JVM)**



JVM

|  |
| --- |
| JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.  JVMs are available for many hardware and software platforms. JVM, JRE, and JDK are platform dependent because the configuration of each [OS](https://www.javatpoint.com/os-tutorial) is different from each other. However, Java is platform independent. There are three notions of the JVM: *specification*, *implementation*, and *instance*.  The JVM performs the following main tasks:   * Loads code * Verifies code * Executes code * Provides runtime environment   JRE  JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used 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.  The implementation of JVM is also actively released by other companies besides Sun Micro Systems.  JRE  JDK  JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and [applets](https://www.javatpoint.com/java-applet). 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.  JDK |