

Difference Between Java and Core Java

The main difference between Java and Core Java is that Java is a programming language that is used for developing applications and applets. Core Java is the basic and raw form of Java which is used for designing application software for desktop as well as server environments. Let us look at the difference between Java and Core Java listed in the table below.

| Java VS Core Java | |
|--|---|
| Java | Core Java |
| Beginners can learn Java because it simply requires the fundamentals. | Object-oriented programming, multithreading, polymorphism, and other concepts are all part of the Java core. |
| For devices that are compatible with Java, it uses a cross-platform approach. | Core Java also uses Java as a computing platform, which is used to create Java applications for both servers and desktops. |
| It is a superset of Core Java. | It is a subset of Java. |
| It is a type of programming language. | It is the most fundamental and standard version of the Java language, and it serves as the foundation for all other Java iterations and editions. |
| It has the ability to lower total network use. As a result, it is incredibly cost-effective. | It is unable to reduce overall network utilization. |
| It's used to create applets and programs. | It is only used to create desktop and server applications. |

What is Java?

Java is an object-oriented, class-based high-level programming language that was specifically designed to have fewer implementation requirements than earlier languages. The V language has affected it, and most of its grammar is borrowed from C and C++. A class file containing Java bytecode can be executed on any platform running a Java Virtual Machine (JVM), regardless of the computer

architecture. It is completely platform agnostic, meaning that it can run any compiled code on any platform.

Java is preferred by programmers and developers above other programming languages for app development due to the amount of knowledge and its growing popularity. The finest part is that it can operate on virtually any device without recompilation, hence the motto "Write once, run anywhere." We can also learn the [Difference Between Java and JavaScript](#).

What is Core Java?

Instead of just the programming language, Core Java refers to a set of libraries. It is the purest form of Java, designed particularly for desktop application development. Simply said, it refers to a subset of Java SE technology that includes both general-purpose and special-purpose APIs.

Core Java is a subset of Java, which contains all of Java's fundamentals, as well as some concepts and package details. It's a self-contained Java application that covers everything from OOP ideas to special operators, data types to wrapper classes, Linked lists to Array lists, queues to exception handling. It is still the most generally used platform for developing portable desktop applications, and it is based on the OOP idea. It includes development tools, a virtual machine, and other class libraries in addition to the general-purpose APIs.

Key Difference Between Java and Core Java

The main difference between Java and Core Java is that Java is a programming language that is used for developing applications. Core Java is the basic and raw form of Java that is used for designing application software for desktop as well as server environments. The key difference between Java and Core Java is provided below.

- Java is intended to have fewer implementation dependencies than other programming languages and supports cross-platform compatibility. Core Java is a programming language that is used to create desktop and server programs.
- Java is the entry-level programming language for those who want to learn the fundamentals of the Java programming language and gain some familiarity with Java SE. Core Java covers all the essentials for developing Java applications, as well as OOP ideas, multi-threading, handling exceptions, polymorphism, and more.

- Java may reduce network consumption, increasing cost-effectiveness. On the other hand, because of a poorly constructed RESTful API that routes too many calls to other services, Core Java is difficult to reduce network utilization.

