

# Features of Java and its importance

Java was created with the primary goal of creating a portable, secure and simple programming language. In addition to this, Java has some other excellent features that play a major role in its popularity. Java buzzwords are the features of Java. Read more here, [Java Classes in Solapur](#).

Below is a list of some of the most important Java features.

- You can also read about how to use
- Object-Oriented
- You can also download the app.
- Platform independent
- Secured
- Robust
- Architecture neutral
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic

## **You can also read about how to use**

Java is easy to learn and understand. Its syntax is clean, simple and straightforward. Sun Microsystem claims that Java is a simple language to program in because:

Java syntax is based upon C++, making it easier for programmers after C ++).

Java has removed many complex and rarely used features such as explicit pointers, overloading operators, etc.

Automatic Garbage Collection is available in Java.

## **Object-oriented**

Java is an object oriented programming language. Java has no objects. Object-oriented software is a combination of data and behaviors that are incorporated into different types of objects.

Object-oriented Programming (OOPs), a method that simplifies the development and maintenance of software by establishing some rules, is a popular programming style.

The basic concepts of OOPs include:

- Object
- Class
- Inheritance
- Polymorphism

- Abstraction
- Encapsulation

### **Platform Independent**

Java is platform-independent

Java is independent of platforms because it differs from other languages such as C++ and C. Java, on the other hand, is a language that can be written once and run anywhere. Platform is the software or hardware environment that a program runs in.

Platforms can be either software-based or hardware-based. Java is a platform that uses software.

Java is different from other platforms because it's a software platform that runs over hardware platforms. It has two main components:

### **Runtime Environment**

API (Application Programming Interface).

Java code is executable on multiple platforms. For example, Windows Linux Sun Solaris Mac/OS etc. The compiler converts Java code into bytecode. This bytecode can run on different platforms.

### **Secure**

Java's security is its most recognizable feature. We can create virus-free systems with Java. Java is secure because:

- No explicit pointer
- Java Programs run inside a virtual machine sandbox
- Classloader: Classloader is part of Java Runtime Environment, which loads Java classes dynamically into the Java Virtual Machine. It increases security by separating classes imported from the network from the files on the local filesystem.
- Bytecode verifier: This program checks code fragments to ensure that they do not contain illegal code which could violate the rights of objects.
- Security manager: This program determines which resources a particular class has access to, such as the ability to read and write data on the local disk.

Java provides these security features by default. Application developers can also provide some security by using SSL, JAAS and Cryptography.

[Java Course in Solapur](#) is a highly demanded course that helps all prospective students get knowledge in Java development and programming.

### **Robust**

The English meaning of Robust has a strong. Java is robust for the following reasons:

Memory management is very strong.

It is difficult to avoid security issues because there are not enough pointers.

Java offers automatic garbage collection, which runs on the Java Virtual Machine. This allows Java applications to dispose of objects that are no longer being used.

Java has both the exception handling and type checking mechanisms. Java is robust because of these features.

### **Architecture-neutral**

Java is architecture-neutral because it has no implementation-dependent features. For example, the size and shape of primitive types are fixed.

In C, the int data type takes up 2 bytes for 32-bit architectures and 4 bytes for 64-bit. In Java, however, the int data type occupies four bytes for both 64-bit and 32-bit architectures.

You can also download the app.

Java allows you to move the Java bytecode from one platform to another. It does not require any implementation.

### **High-performance**

Java is faster than traditional interpreted languages, because Java bytecode "closes" to native code. Although it is a bit slower than a compiled language, such as C ++). Java is an interpretive language, which is why it's slower than other compiled languages like C++ and C.

### **Distributed**

Java is distributed, because it allows users to create Java distributed applications. RMI and EJB can be used to create distributed applications. Java's feature allows us to access files from anywhere on the Internet by calling methods.

### **Multi-threaded**

A thread is a program that runs simultaneously. By defining multiple threads, we can create Java programs that perform many tasks simultaneously. Multi-threading has the main advantage that each thread doesn't take up memory. It uses a shared memory space. Threads are essential for multimedia, Web applications, and other multi-media.

### **Dynamic**

Java is dynamic. Java supports dynamic loading of class. Classes are loaded as needed. It supports native functions, such as C and C++.

Java supports dynamic compilation (garbage management) and automatic memory management.

You can visit here for more information, [Java Training in Solapur](#).