Java Features

Back to: Java Tutorials For Beginners and Professionals

Features of Java Language | Why Java Becomes Popular?

In this article, I am going to discuss **Java Language Features or why Java Becomes so much popular** among developers. At the end of this article, you will understand why java becomes the number one choice for developers and the difference between C, C++, and Java.

Why java is so much more popular now a day's?

Java is so much more popular now a day's because of the following reasons.

Java is simple and familiar:

Java is simple because java simplifies the programmer's job by avoiding certain features of c and C++. Java avoids explicit memory management. Memory management in java is automatic. It is done by JVM. So there are no chances of memory leakage There is no pointer in java. Java is familiar because it is modeled on the Cand C++ Language. Java uses many features of C and C++, therefore java codes look like C++ codes. We can say java is a simplified version of C++.

Java is Portable:

Portability allows the programmer to write the same code for different machines (operating systems). Java provides portability in two ways

1. Source code portability
2. Byte code portability

Java is architecturally neutral:

The behavior of java programs doesn’t change when we move from one system to another which means it will provide the same output in every machine because in java memory layout decisions are not made at compile time, it is made at run time by JVM. But in the case of C and C++ programs, the behavior changes when we move the program from one system to another.

Java is Secured:

Java is secured because of the two reasons

1. There is no pointer in java. A pointer in java is called a restricted pointer means there is no Pointer Arithmetic.
2. Java programs are executed in a secured environment called JVM. JVM will provide security to the java programs.

Java is Robust:

Robust means strong. Java is a strong Type Checking Language having strict Type Checking during both compilation time and execution time which allows us to develop both error-free applications and programs.

Java is Strongly-Typed:

The types of the used variables must be pre-defined and conversion to other objects is relatively strict.

Java is Distributed:

Java provides a set of APIs which allows users to develop distributed applications. It means java language is used for developing distributed applications whose resources are shared by more than one client. An application or program whose resources are shared by more than one client is called a distributed application. As java is used for developing such applications. So, java is a distributed programming language.

Java is Multithreaded:

A process is divided into several small parts known as threads or lightweight processes. Sending multiple threads to the processor for processing is known as Multi-Threading. Multi-threading means handling multiple tasks simultaneously. For example, we can listen to music while scrolling a page and at the same time we can download an application from the internet on a computer. Java supports multithreaded programming.

Java is Dynamic:

Linking of the program is of two types

1. Static linking
2. Dynamic linking

Static linking: linking of all executable blocks is done before executing programs. If there is a small change in the executable block need to compile the whole program. So here wastage of memory and the efficiency of the whole program decreased. Example: C and C++.

Dynamic linking: Loading and linking of all executable blocks done at the time of program execution is called dynamic linking. It increases the efficiency of the program. Any small change in the executable block, need not compile the whole program. So there is no waste of memory in java.

OR:

The link between the function call and function definition is called binding. There are two types of binding

1. Static binding
2. Dynamic binding

When a method is invoked, if its method definition which is bound at compilation time by the compiler is only executed by JVM at run time, then it is called compile-time polymorphism or static binding, or early binding.

When a method is invoked, the method definition which is bound at compilation time by the compiler is not executed at run time by JVM, instead, it is executed from the subclass based on the object stored in the referenced variable called run time polymorphism/dynamic binding. So java is dynamic.

Java is Compiled and Interpreted:

We know a programming language is either compiled or interpreted. But java combines both approaches. That's why java is called a two-stage system.

First java compiler JAVAC translates source code into an intermediate code known as byte code. But codes are not machine instructions. So in the second stage, this byte code is interpreted by the java interpreter (JVM) in java. As a result machine instructions will be generated which are directly executed by the machine. Hence we are saying java is both interpreted and compiled language.

Java is Object-Oriented:

Except for the primitive data types, all elements in Java are objects. Object-oriented is not a programming language, it is a programming technique or concept, or principle which defines a set of rules and regulations for organizing the data and instructions. The concepts provided by oops are as follows

1. Encapsulation
2. Abstraction
3. Polymorphism
4. Inheritance

The programming language supports these four features known as an object-oriented programming language. Java supports these four features so java is object-oriented.

Java is Platform Independent :

Many languages are compatible with only one platform. Java was specifically designed so that it would run on any computer, regardless if it was running Windows, Linux, Mac, Unix, or any of the other operating systems.

Automatic Memory Management :

Java manages memory allocation and de-allocation for creating new objects. The program does not have direct access to memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.

Java definition based on these features:

Java is just a simple, secure, robust, portable, platform-independent, architectural neutral, multithreaded, object-oriented programming language with a strong type exception handling mechanism for developing distributed, dynamically extensible programs

What is the Abbreviation of Java?

There is no abbreviation of java. The development team of java just has chosen this name. The name java specifically doesn't have any meaning rather it refers to the hot, aromatic drink coffee. This is the reason the java programming language is a coffee cup.

Difference Between C, C++, and JAVA

JAVA	C	C++
Java is a pure object-oriented language.	C is procedural language.	C++ is an object-oriented language.
We can create our own package in Java.	Cannot create our own package in C.	Cannot create our own package in C++.
Internet programming like Frame and Applet is used in Java.	Frame and Applets are not used in C.	Frame and Applets are not used in C++.
Java uses a Compiler and Interpreter.	C uses only the Compiler.	C++ also uses only the Compiler.
Multiple Inheritance is not used in Java.	There is no Inheritance	Multiple Inheritance is used in C++.

Instead of multiple inheritance Java uses Interfaces.	in C.	
Java is a platform-independent language.	C is platform-dependent.	C++ is also platform-dependent.
Java doesn’t support any header files.	In C, we use stdio.h header file.	In C++, we use iostream.h and conio.h header files.
Java does not support Pointers.	Pointers are used in C.	Pointers are used in C++.
Exception Handling is supported by Java.	There is no Exception Handling in C.	C++ supports Exception Handling.
Java is used to develop Android Applications.	C is used in Embedded Programming.	C++ is used in Game Development and System Programming.

Terminology used in Programming Language:

1. **Source code:** The developer-written program is nothing but the source code. It is written according to the programming language syntax.
2. **Compiled code**: a compiler-generated program that is converted from source code is called compiled code.
3. **Compiler:** it is a translation program that converts the source into machine language at once.
4. **Interpreter**: it is also a translation program that converts source code into machine language but line by line.
5. **Executable code**: OS understandable executable programs (.exe files)
6. **Compilation:** it is a process of translating source code into compiled code.
7. **Execution:** it is the process of running the compiled code to get the output.

In the next article, I am going to discuss **Java Application Development Lifecycle** in detail i.e. I am going to discuss three important concepts i.e. **JDK, JVM, and JRE** in detail. Here, in this article, I try to explain the **Java Language Features and discuss why Java is so much popular among developers** nowadays.

DOT NET TUTORIALS

Dot Net Tutorials

About the Author: Pranaya Rout

Pranaya Rout has published more than 3,000 articles in his 11-year career. Pranaya Rout has very good experience with Microsoft Technologies, Including C#, VB, ASP.NET MVC, ASP.NET Web API, EF, EF Core, ADO.NET, LINQ, SQL Server, MYSQL, Oracle, ASP.NET Core, Cloud Computing, Microservices, Design Patterns and still learning new technologies.

Back to Course

Java Tutorials For Beginners and Professionals

Next Lesson

Java Application Development Lifecycle

1 thought on “Java Features”

SANJAY KANT

AUGUST 22, 2025 AT 9:49 AM

Java is an object-oriented language, but not a purely object-oriented one.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name*

Email*

Website

Post Comment

Java Basics

- **Java Features**
- Java Application Development Lifecycle
- Environment Setup for Java
- Creating First Java Program
- Data Types in Java
- Literals in Java
- Type Casting in Java
- Operators in Java
- Variables in Java
- Identifiers and Reserved Words in Java
- Control Flow Statements in Java
- Looping Statements in Java
- Branching Statements in Java
- Methods in Java
- Java User Input and Output
- Pass By Value and Pass By Reference in Java
- Command Line Arguments in Java

Java OOPs

- Object-Oriented Programming in Java
- Class and Objects in Java
- Constructors in Java
- Inner Classes in Java
- Wrapper Classes in Java
- Polymorphism in Java

- Encapsulation in Java
- Access Modifiers in Java
- Inheritance in Java
- Abstraction in Java
- Abstract Classes and Abstract Methods in Java
- Interface in Java
- Association Composition and Aggregation in Java
- Garbage Collection in Java
- Final Keyword in Java
- Static Keyword in Java

Java Exception Handling

- Exception Handling in Java
- Finally Block in Java
- throw and throws keywords in Java
- Custom Exception in Java
- Exception Propagation in Java

Java Strings, Packages, JVM & IO Streams

- String in Java
- Java Packages
- JVM Architecture
- Java IO Stream
- Byte Streams in Java
- Character Streams in Java
- Serialization and Deserialization in Java

Java Array, Collection & Generics

- Array in Java
- Multi Dimensional Arrays in Java
- Java Collections Framework
- List Collections in Java
- Cursors of Collection Framework in Java
- Set Collections in Java
- Queue Collections in Java
- Map Collections in Java
- Sorting Collections in Java
- Generics in Java

Java Multithreading

- Multithreading in Java

- Thread Class in Java
- Thread Life Cycle in Java
- Thread Priority in Java
- Daemon Thread in Java
- Thread Synchronization in Java
- Inter Thread Communication in Java
- Deadlock in Java
- Multithreading Exercises in Java

Java Applet, AWT & Event Handling

- Applet in Java
- Graphics in Applet
- Abstract Windows Toolkit (AWT) in Java
- AWT Controls in Java
- Event Handling in Java
- Event Listener Interfaces in Java
- Layout Manager in Java

Java Swings

- Swings in Java
- Swing Controls in Java
- Swing Dialog Box in Java
- Working with Image Menus and files in Java Swings
- Working with Tables and Progress Bars in Java Swings

Java Enumeration

- Enumeration in Java

JDBC in Java

- JDBC in Java
- JDBC Architecture
- JDBC Drivers
- Steps to Design JDBC Applications in Java
- CRUD Operations in Java using JDBC
- JDBC ResultSet
- Prepared Statement in JDBC

Java Advanced Features

- Internationalization in Java
- Regular Expression in Java
- Parallel Programming in Java
- Reflection in Java

- ✓ Date and Time API in Java
- ✓ Java Calendar Class
- ✓ Java System.exit() Method
- ✓ How to work with JSON in Java

Java Interview Questions

- ✓ Java Interview Questions and Answers

Java Popular Books

- ✓ Most Recommended Java Books
- ✓ Most Recommended Data Structure and Algorithms Books using Java
- ✓ Most Recommended JDBC Books
- ✓ Most Recommended Hibernate Books

Games

- ✓ Ranked: Easiest and Toughest Card Games to Program

About Us Privacy Policy Contact ADO.NET Tutorial Angular Tutorials ASP.NET Core Blazor Tuturials
ASP.NET Core Tutorials ASP.NET MVC Tutorials ASP.NET Web API Tutorials C Tutorials
C#.NET Programs Tutorials C#.NET Tutorials Cloud Computing Tutorials
Data Structures and Algorithms Tutorials Design Patterns Tutorials DotNet Interview Questions and Answers
Core Java Tutorials Entity Framework Tutorials JavaScript Tutorials LINQ Tutorials Python Tutorials
SOLID Principles Tutorials SQL Server Tutorials Trading Tutorials JDBC Tutorials Java Servlets Tutorials
Java Struts Tutorials C++ Tutorials JSP Tutorials MySQL Tutorials Oracle Tutorials
ASP.NET Core Web API Tutorials HTML Tutorials