Java

**What is Java?**

 Java is a cross-platform object-oriented programming language that was released by Sun Microsystems in the year 1995. Today, Java is needed to run various applications such as games, social media applications, audio and video applications, etc.

**What is Java used for?**

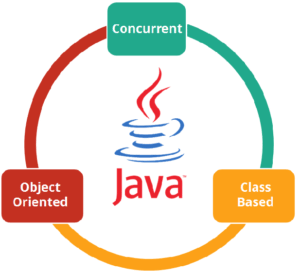
Before I go ahead with this, let me brief you about why you should choose Java. It is highly popular and has dominated this field from early 2000’s till the present 2018.

Some of the applications are listed below:

* **Banking**: To deal with transaction management.
* **Retail**: Billing applications that you see in a store/restaurant are completely written in Java.
* **Information Technology**: Java is designed to solve implementation dependencies.
* **Android**: Applications are either written in Java or use Java API.
* **Financial services**: It is used in server-side applications.
* **Stock market**: To write algorithms as to which company they should invest in.
* **Big Data**: Hadoop MapReduce framework is written using Java.
* **Scientific and Research Community**: To deal with huge amount of data.

**What is Java?**

It is an object-oriented language similar to C++, but with advanced and simplified features.This language is **free to access** and can **run** on **all platforms**.



Java is: –

* **Concurrent** where you can execute many statements instead of sequentially executing it.
* **Class-based** and an **object-oriented** programming language.
* **Independent** programming language that follows the logic of “**Write once, Run anywhere**” i.e. the compiled code can run on all platforms which supports java.

In simple words, it is a computing platform where you can develop applications.

## ****Features Of Java****

**Simple:**Java has made life easier by removing all the complexities such as pointers, operator overloading as you see in C++ or any other programming language.

**Portable:**This is platform independent which means that any application written on one platform can be easily ported to another platform.

**Object-oriented:**Everything is considered to be an “**object**” which possess some state, behavior and all the operations are performed using these objects.

**Secured:**All the code is converted in **bytecode** after compilation, which is not readable by a human. and java does not use an explicit pointer and run the programs inside the sandbox to prevent any activities from untrusted sources. It enables to develop virus-free, tamper-free systems/applications.

**Dynamic:**It has the ability to adapt to an evolving environment which supports dynamic memory allocation due to which memory wastage is reduced and performance of the application is increased.

**Distributed:**This language provides a feature which helps to create distributed applications. Using Remote Method Invocation (RMI), a program can invoke a method of another program across a network and get the output. You can access files by calling the methods from any machine on the internet.

**Robust:**Java has a strong memory management system. It helps in eliminating error as it checks the code during compile and runtime.

**High Performance:**Java achieves high performance through the use of bytecode which can be easily translated into native machine code. With the use of JIT (Just-In-Time) compilers, it enables high performance.

**Interpreted:** Java is compiled to bytecodes, which are interpreted by a run-time environment.

**Multithreaded:** Java supports multiple threads of execution (a.k.a., lightweight processes), including a set of synchronization primitives. This makes programming with threads much easier.

## ****Components****

**JVM (Java Virtual Machine)**

It is an abstract machine. It is a specification that provides a run-time environment in which the bytecode can be executed. It follows three notations:

* **Specification**: It is a document that describes the implementation of the JVM. It is provided by Sun and other companies.
* **Implementation**: It is a program that meets the requirements of JVM specification.
* **Runtime Instance**: An instance of JVM is created whenever you write a command on the command prompt and run the class.

**JRE (Java Runtime Environment)**

JRE refers to a runtime environment in which bytecode can be executed. It implements the JVM and provides all the class libraries and other support files that JVM uses at runtime. So JRE is a software package that contains what is required to run a program. Basically, it’s an implementation of the JVM which physically exists.

**JDK(Java Development Kit)**

It is the tool necessary to:-

* Compile
* Document
* Package Java programs.

The JDK completely includes JRE which contains tools for programmers. The Development Kit is provided free of charge. Along with JRE, it includes an interpreter/loader, a compiler (javac), an archiver (jar), a documentation generator and other tools needed in Java development. In short, it contains JRE + development tools.

**Why is Java so popular?**

The principle of Java, WORA, makes it an eye-catching language! Imagine you have created a code and you can run it on any machine and work anywhere you want. Such a privilege and ease! This is what attracts the programmers the most. [Java programs](https://www.edureka.co/blog/java-programs/) are capable to execute on different machines having JRE. JRE is compatible with all the devices, say, mobile phones, PC’s, Linux, Windows, etc.

Moving on with the next segment.

**Where is Java used?**

Java has been the choice of fresher as well as experienced programmers.

* It is used in designing web applications
* It is widely used in creating [Android applications](https://www.edureka.co/blog/java-for-android/)
* Java is also a great choice for the developers in creating scientific applications
* It used in developing software tools too
* Game development

**Top 10 Reasons To Learn Java**

1. [Java’s Popularity & High Salary](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#Java'sPopularity)
2. [Java is Portable & Versatile](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#Portable)
3. [Java is an Object Oriented Programming Language](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaOOPs)
4. [Demand: Java is everywhere](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#Demand)
5. [Java Development Tools](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaDevelopmentTools)
6. [Java Applications](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaApplications)
7. [Tons of resources & Community Support](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaCommunity)
8. [Java EE & its rich API](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaAPI)
9. [Java New Features](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaFeatures)
10. [Java is easy & Open Source](https://www.edureka.co/blog/top-10-reasons-to-learn-java/#JavaOpenSource)

New Keyword

[Java New Keyword - Javatpoint](https://www.javatpoint.com/new-keyword-in-java)

# **Can we declare a constructor as private in Java?**

[Can we declare a constructor as private in Java? (tutorialspoint.com)](https://www.tutorialspoint.com/can-we-declare-a-constructor-as-private-in-java#:~:text=Yes%2C%20we%20can%20declare%20a,in%20the%20Singleton%20Design%20Pattern.)

**Exception Handling** Diagram

Description automatically generated

## Difference between Checked and Unchecked Exceptions

### **1) Checked Exception**

The classes that directly inherit the Throwable class except RuntimeException and Error are known as checked exceptions. For example, IOException, SQLException, etc. Checked exceptions are checked at compile-time.

### **2) Unchecked Exception**

The classes that inherit the RuntimeException are known as unchecked exceptions. For example, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException, etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

### **3) Error**

Error is irrecoverable. Some example of errors are OutOfMemoryError, VirtualMachineError, AssertionError etc.

## Java Exception Keywords

Java provides five keywords that are used to handle the exception. The following table describes each.

|  |  |
| --- | --- |
| **Keyword** | **Description** |
| try | The "try" keyword is used to specify a block where we should place an exception code. It means we can't use try block alone. The try block must be followed by either catch or finally. |
| catch | The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later. |
| finally | The "finally" block is used to execute the necessary code of the program. It is executed whether an exception is handled or not. |
| throw | The "throw" keyword is used to throw an exception. |
| throws | The "throws" keyword is used to declare exceptions. It specifies that there may occur an exception in the method. It doesn't throw an exception. It is always used with method signature. |