

# **65+ Most Asked JAVA Interview Questions & Answers**

## **Java Interview Questions for Freshers & Beginners**

- **What is Java? Explain its meaning and definition.**

It is one of the most popular programming languages today, with its role in the development of web applications, mobile apps, software, gaming systems, as well as server-side technologies.

Java programming has been here for over 20 years now and has been nothing less than a boon for programmers and developers. Famous as an object-oriented programming language, Java is not only multi-purpose, but also a secure, high-performing, and trusted coding language. This is the reason behind its usage in even enterprise-grade software development.

- **The syntax of Java is based on which programming language?**

C and C++ programming are the base of the Java syntax.

- **How is Java platform independent?**

Among the list of Java interview questions, you can be asked why Java is called platform independent. You must be ready with the correct answer.

The meaning of platform-independent is that you can write a Java program on one machine and execute it on other machines or platforms. It has become possible because of the bytecode, and VMs can handle this code accordingly. So, there will be no issues related to the hardware when it comes to running the code.

- **When was Java developed?**

Java was developed in the year 1991.

- **Who developed the Java programming language?**

Java was developed by James Gosling. He was a computer scientist based in Canada, and is famous as the founder of Java programming. When he invented Java, he was working at Sun Microsystems, which Oracle later acquired.

- **What does 'write once run anywhere' mean in Java?**

Write once, run anywhere, or WORA in Java means that it is a coding language where you write a program for some purpose only once and then use it or run it across multiple operating systems. For instance, you can write a program and run it on Windows, macOS, Android, Linux, etc.

"Java's write once, run anywhere" term was first initiated by Sun Microsystems, where the founder of this language used to work. This characteristic makes Java a portable programming language.

- **What is Java programming used for? Explain its primary applications.**

There is a wide range of use cases of Java programming language. Below are its main applications:

- **Mobile App Development**

Despite the introduction of Kotlin, Java is still used as a reliable programming language for Android app development. This coding language has the software development kits (SDKs) and libraries that are required to develop mobile apps.

- **Chatbot Development**

Another use of Java is in chatbot development. Smart chatbots that use natural language processing (NLP) can be built using this programming language.

- **Development of Games**

One of the most important applications of Java is in building games or gaming apps. Some of the world-famous games like Minecraft, Spiral Knights, SimCity, Saints Row 2, Asphalt 3, FIFA 11, Wakfu, Tokyo City Nights and many more are built on Java.

- Cloud Computing

The write once, use anywhere characteristic of Java makes it a highly applicable language for cloud applications as well. Plenty of cloud platforms rely on this programming for a decentralized experience.

- Big Data

Big data platforms or tools heavily depend on Java, and it is considered a language on which the future of big data relies. This is because of its features that enable faster processing of large sets of data.

- Enterprise-grade Web Apps

Enterprise-level apps that are mission-critical are developed using Java programming. Even top brands like Wipro, Google, Infosys, and HCL use it to develop enterprise apps. It is because of its high performance it enables and supports a wide range of server-side technologies.

Some of the most popular web apps built on Java include LinkedIn, IRCTC, and AliExpress.

- Internet of Things (IoT)

In IoT technology, sensors and hardware devices process the data. These things are mostly run using programs written in Java programming language.

- Artificial Intelligence (AI)

As one of the most suitable programming languages for artificial intelligence (AI) projects, Java can be used to develop intelligent solutions. For instance, it is great for building search algorithms, neural networks, ML-based services, deep learning applications, etc.

- **What is Java Virtual Machine (JVM)?**

JVM in Java, as the name suggests, is a virtual machine that plays a crucial part in the execution of source code. It works as an abstraction layer between the runtime environment and the hardware.

- **What is Java Runtime Environment (JRE)?**

JRE in Java is simply an environment that allows developers or programmers to run Java-based apps on operating systems. You can say that it facilitates the interaction between OS and the program.

Java JRE provides several resources to programmers, such as libraries, JVM, Java Plug-in, Web Start, etc. It is available to download on Windows, Linux, macOS, and Oracle Solaris.

- **What is Java SE (Standard Edition)?**

Java Standard Edition, abbreviated as Java SE, is a computing platform on which programmers and developers build and deploy Java-based projects. This platform comes with plenty of Java libraries and APIs, including `java.util`, `java.net`, `java.math`, `java.io`, and many more.

- **What are operators in Java?**

Java operators are simply the symbols used to perform a wide range of different operations. Every operator has its specific operation or functionality.

For instance, you can use the `+` operator for the addition of two values, the `-` operator for subtraction, the `*` operator for multiplication, and the `/` operator for division.

- **What are the different types of Java operators?**

You can classify the operators in Java into five categories, as mentioned below:

- Arithmetic operators

These are used for mathematical calculations, or arithmetic operations, to be precise.

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo

- Unary operators

These operators play a vital role in performing operations related to increasing or decreasing the values.

Unary minus (-)	To make a value negative
Unary plus (+)	Generally not used as values are positive by default
Increment (++)	To increase the value by 1
Decrement (--)	To decrease the value by 1

Inverting (!)	To inverse the value
---------------	----------------------

- Assignment operators

These operators, as the name suggests, are used to assign values to variables in a Java program.

Operator	Presentation	Meaning
=	X = Y	X = Y
+=	X += Y	X = X + Y
-=	X -= Y	X = X - Y
*=	X *= Y	X = X * Y
%=	X %=	X = X % Y

- Relational or comparison operators

If you want to see the relations between values, then the relational operators in Java are used. For instance, you can check whether the given values are equal to each other, greater/lesser than each other, greater than equal to, or lesser than equal to each other.

If the boolean values don't meet the relational operator criteria, then accordingly, it returns *true* or *false*.

Operator	Meaning
==	Is equal to
!=	Is not equal to

>	Is greater than
<	Is lesser than
>=	Is greater than or equal to
<=	Is lesser than or equal to

- Logical operators

The digital electronics field heavily depends on AND and OR gate operations. The same applies in Java for decision-making based on logical AND and OR operators.

Operator	Meaning
&&	Logical AND

	Logical OR
!	Logical NOT

- **What is JIT compiler in Java?**

Java JIT compiler or Just-In-Time compiler is one of the key parts of Java Runtime Environment (JRE). Its role is the compilation of bytecode to native machine code when it's in runtime. As a result, the overall performance of the Java apps is optimized.

- **What is a Java class?**

A class in Java is a template based on some logic using which we can create multiple objects that follow the same logic as the class. The role of a Java class is to define the data types and methods of the objects.

In simple terms, you can call the class the main category, which includes several items called objects. Here, the objects under a class will have similar properties or characteristics.

**For example**, if you visit Amazon and browse the “Mobiles” category, it will show you smartphones of all brands and types. But all these smartphones will have similar properties like a camera, some RAM, the ability to make calls, download apps, send messages, etc.

So, here, the ‘Mobiles’ is a class, and all the smartphones are objects.

- **What is a package in Java?**

A package in Java is like a folder that holds classes, interfaces, and sub-packages. Here, the point to be noted is that everything in this folder or Java package has some similarity or relation based on their functionality.

The aim of using a Java package is to better organize the workloads, avoid conflicts in names, and control access.

**In Java, there are two types of packages:**

- Built-in packages
- User-defined packages

The built-in packages include java.lang, java.util, java.io, and java.net, which can be used from Java API. Apart from these, you can also build your own packages, which are called user-defined packages.

- **What are keywords in Java?**

Keywords in Java programming are actually some predefined words in syntax that a programmer can't use in the form of classes, methods, identifiers, or variables. These are also known as reserved words in Java.

- **How many keywords are there in Java?**

There are over 50 keywords in Java. Here is the full list in alphabetical order:

- abstract
- assert
- boolean
- break
- byte
- case
- catch
- char
- class
- continue
- const
- default
- do
- double
- else
- enum
- exports
- extends
- final
- finally
- float
- for
- goto
- if
- implements
- import
  
- instanceof
- int
- interface
- long
- module
- native
- new
- package
- private
- protected

- public
- requires
- return
- short
- static
- strictfp
- super
- switch
- synchronized
- this
- throw
- throws
- transient
- try
- var
- void
- volatile
- while

- **What are the key features of Java?**

It is one of the basic Java interview questions for freshers, and sometimes for experienced professionals as well.

Here are the top 10 features of Java that you must know:

- Simple, Clean, Easy to Learn

One of the best things about Java is that it is easy to learn and understand, even for beginners. Its syntax is simple as it is based on basic languages like C++. The code written in Java is also clean and easy to run.

- Object-oriented Programming Language

Java is completely based on objects, hence it is called an object-oriented programming language.

- Java is Both Compiled and Interpreted

A programming language is generally compiled or it is interpreted. Only rare languages exhibit both things. Java is one such coding language that has the features of both compilation and interpretation.

- Java is Platform Independent



This is one of the top features of Java programming. The meaning of platform-independent here is that you can write a Java program on one machine and execute it on other machines or platforms. It has become possible because of the BYTE code.

- Portability

The portability in Java is the result of having features like platform independence and architecture neutrality. Programmers can run Java code on a wide range of virtual machines and hardware because its bytecode can be converted accordingly.

- Robust Programming Language

The abilities of Java like garbage collection and exception handling make it a solid programming language.

- Highly Secure

Security plays a crucial part whether you are developing a basic app or a business-critical solution. On that front, Java is considered as the most secure language as it helps in writing code that is free from viruses and other security threats. That is the reason behind its application in enterprise-grade app development.

- Java Multithreading Features

The multithreading feature helps in writing code that can perform multiple tasks simultaneously. Moreover, the thread tasks consume less processing power and memory.

- Easy Interpretation

Regardless of the computer architecture, Java programs can be run and interpreted on any type of machine. You can call it architecture-neutral language.

- High Performance

Instead of being an interpreted language, Java offers faster performance because of its just-in-time compiler.

- **What is object in Java?**

As you might already know that Java is all about classes and objects. But what is an object? This is a crucial topic that you should include in your list of Java interview questions and answers for freshers. Let's know its meaning below.

An object is an instance of a class in Java. In fact, it is created from a class only using the 'new' keyword. Every object has its identity, behavior, and state, the way things in the real world also have these three things.

- **What is difference between Java and JavaScript?**

There are several differences between Java and JavaScript. Whether you are a fresher or an experienced professional, this is among the top Java interview questions for you. Below, we have curated a tabular comparison of Java vs JavaScript so that it becomes easier for you to understand the main differences.

Java	JavaScript
Object-oriented programming language	Object-based scripting language
Can be used for complicated tasks and processes	Can't be used for complicated tasks
Needs code compilation	Text-based code
Independent language	Needs to be used with HTML
Strongly typed programming language. Need to declare variables before using them in the program.	Loosely typed language. No issues whether data types are declared or not
It's statically-type	It's dynamically-typed
High memory consumption	Low memory consumption
Saved as byte code	Saved as source code
For concurrency, it uses threads	For concurrency, it uses events
.java extension used to save programs	.js extension used to save programs
Supports multithreading	Doesn't support multithreading
Objects are based on class	Objects are based on prototype
Need JDK or Java Development Kit to run the code	Need text editor to run the code
Primarily used for backend development	Can use for both front-end and back-end

- **Which Java class is considered a superclass of all other classes?**

The object class is considered the superclass of all the remaining classes.

- **Is it possible for a class to extend itself?**

No. It's not possible.

- **What is difference between Java and C++ programming?**

Another important Java interview questions for freshers and experienced developers can be about the differences between Java and C++ programming languages. To help you understand it easily, we have created the following comparison between the both:

Java	C++
Platform independent	Platform dependent
Uses compiler and interpreter both	Compiler only
Garbage controller to automate memory management	Manual memory management
Support for comments	Doesn't support comments
Doesn't support goto statement	Supports goto statement
Developed by James Gosling	Developed by Bjarne Stroustrup
Used for various types of development purposes, like web apps, Windows apps, etc.	System programming is the primary use.
Supports procedural and object-oriented programming both	Supports object-oriented programming only
Limited number of libraries	Large number of libraries
Write once, run anywhere	Write once, compile anywhere

- **Explain the difference between JDK, JRE, and JRM.**

Java interview questions about the difference between Java JDK, JRE, and JRM are very common. Here is the tabular comparison to help you find the right answer.

JDK	JRE	JVM
Java Development Kit	Java Runtime Environment	Java Virtual Machine
An SDK required to build Java-based apps. It comes with several tools like debugger, compiler, and more.	A software package that comes with class libraries. Used for running Java projects.	A virtual machine that makes Java a platform-independent language.
Platform-dependent.	Platform-dependent.	Platform-independent.
Mostly used in code execution during development.	Used for providing an environment where the code execution can be done.	Used to define the execution and supporting JRE.
JDK = JRE + Development	JRE = JVM + Libraries	JVM = Support JRE to load,

Tools		verify, and execute code
Comes with various tools related to debugging, monitoring, and overall development.	Comes with class libraries and supporting files.	Doesn't include any tools or a library.

- **Is it possible to assign a superclass to a subclass in Java?**

Not. It's not possible.

- **How to print text in Java?**

The `println()` and `print()` methods are used to print a text in Java.

- `println()` example

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hi There!");  
        System.out.println("Welcome to WsCube Tech!");  
        System.out.println("Let's know top Java Interview Questions and Answers!");  
    }  
}
```

**Output:**

Hi There!

Welcome to WsCube Tech!

Let's know the top Java Interview Questions and Answers!

- `print()` example

```
public class Main {  
    public static void main(String[] args) {  
        System.out.print("Hi There! ");  
        System.out.print("I can now learn the core interview questions on Java.");  
    }  
}
```

**Output:**

Hi There! I can now learn the core interview questions on Java.

## **Intermediate Java Interview Questions for Experienced (2-5 Years)**

- **What is multithreading in Java?**

When a Java program is divided into multiple small parts, and these parts are executed parallelly and run simultaneously, this process is called multithreading.

The role of Java multithreading is to create lightweight programs or threads so that processing power can be used in an optimum manner.

- **Which class is used for multithreading in Java?**

The Java Thread class is used to execute multithreading. It enables the creation of small threads which can run in a concurrent manner.

- **What is Java applet?**

An applet in Java is used for setting up dynamic content on a web page. You can call it a program that is added to the page, which then runs in a browser and shows the dynamic content to the end user.

Some of the primary benefits of a Java applet include lower response time at the front end, secure code, and the applet program working great on popular operating systems, like Windows, Linux, and macOS.

- **What is garbage collection in Java?**

It is an important concept that you must know while appearing for the interview. This is one of the most asked Java garbage collection interview questions.

The meaning of garbage collection in Java programming is that it automates the memory management for programs running on the Java virtual machine.

So, when you create and run programs, heaps of memory are allocated for memory consumption. In the long run, there will be several objects of the Java program that won't be required. What the garbage collection does is remove the unused objects automatically. As a result, it optimizes memory.

- **What if you use Java keywords as a variable or identifier?**

In case a program contains keywords as a variable, class, etc., then it will show a compile-time error.

- **What is inheritance in Java?**

If you have been searching for Java OOPS interview questions and answers, then this is going to be a top question. It is because inheritance is a crucial concept in Java related to object-oriented programming (OOP).

In simple terms, Java inheritance means creating classes that inherit features of some other classes. So, if you want to build a class that has some relationship with any other classes, then you need to use the inheritance method.

As a result, the new class that you create will have the features of the inherited class. It is a good mechanism to use to set up a hierarchy between classes.

- **What is polymorphism in Java?**

This is yet another important concept in the list of top Java interview questions and answers related to OOPS.

The general meaning of polymorphism is “the condition of occurring in several different forms.” So, polymorphism in Java can be defined as the ability of a class to take different forms or to provide different functionalities.

For example, you can use this method to show a single message in multiple forms, based on the set parameters.

- **What is encapsulation in Java?**

If you have been in the field of programming for some time now, then one of the core Java interview questions for 3 years experience will be about encapsulation. Let's understand its meaning here.

Java encapsulation is a process that allows you to integrate the data variables and code and store them as a single thing. It is like two different capsules of medicine are mixed together to create a new single capsule. This is the logic behind calling it encapsulation.

An important thing to know here is that after the encapsulation of variables of a class, It won't be possible for other classes to access these variables.

- **What is serialization in Java?**

Java interview questions related to serialization are so common to be asked when you appear for a job interview.

When there is a need for transferring an object code in Java from one JVM to another, then the serialization process is used. Here, what it does is convert the object code to a stream so that it can be transferred to another JVM over the internet or network.

Once the stream is received on another JVM, it then goes through the deserialization process so that it can be converted back to the object code and brought into use.

- **What is Java JDBC?**

JDBC stands for Java Database Connectivity.

It is an API to establish and manage a connection to the database right from within the Java program. Using JDBC, the developers or programmers can connect to not only one but multiple databases.

You can say that it works like a communication channel between the program and the database. Developers can use it to connect to any database if the relevant drivers are in use.

Since it is not a basic topic, it is an important concept for people looking for core Java interview questions for experienced.

- **What is Java enum?**

Enum in Java stands for enumeration. It is a data type that comes with a set of pre-defined constant values. These values are separated by a comma. The enum concept was brought to this programming as part of Java 5. For declaring the enums, the *enum* keyword is used.

- **What is constructor overloading in Java?**

Another concept to keep in your list of Java interview questions and answers for experienced developers is constructor overloading.

The role of Java constructors is to define the state of an object. When there are various constructors of a single class to be defined, it is called constructor overloading. As a result, a class becomes capable of possessing multiple constructors.

- **What is copy constructor in Java?**

It is a constructor that is used when you need to make a copy of an object of an existing object of the same class. Here, the new copy of the object wouldn't impact the existing object.

- **Which keyword in Java is used to inherit a class?**

We should use the *extends* keyword for this purpose.

- **What are the top benefits of inheritance in Java?**

Following are the main benefits of Java inheritance:

- Code reusability
- Method overriding
- Ability to achieve runtime polymorphism
- Optimize duplicate code
- Improve the redundancy of the app
- Code flexibility so that it can be changed easily

- **What are the different memory areas assigned by JVM?**

There are five types of memory areas that a JVM can allocate:

- Class(Method) Area
- Heap memory
- Stack memory
- Program Counter Register
- Native Method Stack

- **What are access modifiers in Java?**

As a programming enthusiast, you should know about the access modifiers while preparing for the Java interview questions and answers.

As the name suggests, the access modifiers in Java are used to manage the access level for classes, variables, methods, constructors, etc. The access can be changed or specified using these access modifiers.

**Access modifiers are of four types:**

- Public
- Private
- Default
- Protected

- **How many types of inheritance are there in Java?**

There are five types of Java inheritance:

- Single-level inheritance
- Multi-level Inheritance
- Hierarchical Inheritance
- Multiple Inheritance
- Hybrid Inheritance

- **Can you restrict an object from inheriting its subclass? If yes, how?**

Yes. It is possible if we declare the member or object private. In such a case, the subclass can't access the private members directly.

- **How can we remove the duplicate elements from a list of numbers if Java 8 is being used?**



First, we need to apply the stream so those duplicate elements will be found and then make a new collection by applying `Collections.toSet()` method.

## Advanced Java Interview Questions and Answers for Experienced (5-10 Years)

- **What is the difference between heap memory and stack memory in Java?**

Two types of memory are used in the Java Virtual Machine (JVM). One is heap memory, and another is stack memory.

The primary difference between the two is that heap memory's role is to store objects, whereas stack memory stores local variables and the order of method execution. The following tabular comparison shows all the key differences between the both. While preparing for Java interview questions and answers, ensure to understand this concept well.

Heap Memory	Stack Memory
Used to save JRE classes and objects	Used to save methods, variables, and reference variables
Memory size is larger	Memory size is small
It takes more time to access or allocate heap memory	It takes less time to access or allocate stack memory
No fixed format or order	LIFO (Last In First Out) order
Allows changes to the allocated memory	Doesn't allow changes to the allocated memory
-Xmx and -Xms are used to increase/decrease memory size	-Xss is used to increase memory size
Memory allocation or deallocation is done manually	Memory allocation or deallocation is done using compiler
Shared memory for all threads	Dedicated memory for every object
Higher cost	Lower cost

- **Which are the best Java compilers?**

If you are an experienced developer, then you must know about the top Java compilers. Because this is going to be one of the top interview questions on Java for experienced professionals.

**Here is the list of best compilers for Java programming:**

- Eclipse
- NetBeans
- Xcode
- AndroidStudio
- Tabnine
- Codota
- Codenvy
- JDeveloper
- jGrasp
- IntelliJ IDEA
- BlueJ
- MyEclipse
- Slickedit
- JBoss Forge
- JEdit

- **What is the difference between equals() method and equality (==) operator in Java?**

There are a number of key differences between the equals method and the equality operator in Java. The primary difference is that one is a method, and another is an operator.

Such tricky concepts are usually asked when you have some experience in this field. So, you need to study the core Java interview questions and answers for experienced professionals really well.

For this question, we have created a tabular comparison to help you understand the differences between the equals method and equality operator in Java.

Equals() Method	Equality Operator (==)
It is a method	It is an operator
Its role is for comparing the content of an object	Its role is for comparing the reference values and objects
It can be overridden	Can't be overridden
Can't be used with primitives	Can be used with primitives

- **Can you inherit static members to a subclass?**

No. It can't be done.

- **Can you override the final method in Java?**

No. It can't be overridden.

- **How to declare an infinite loop in Java?**

You must be well-prepared for such Java programming interview questions. There are three ways to declare an infinite loop in Java.

- While loop

**Syntax**

```
while(condition){  
    //code  
}
```

- For Loop

**Syntax**

```
for(initialization;condition;updation){  
    //code  
}
```

- Do-While Loop

**Syntax**

```
do{  
    //code  
}while(condition);
```

- **What are the roles of final, finally, and finalize keywords in Java?**

There are 50+ keywords in Java, and three similar-sounding keywords from them are final, finally, and finalize. Let's understand the differences between them with the following comparison. You must know it because it can be one of those core Java interview questions for experienced.

final	finally	finalize
Its role is to execute limitations or restrictions on classes, methods, or variables	Its role is in exception handling. finally keyword runs the crucial code no matter whether the exception occurs or not.	Used for processing clean up during garbage collection.

Can be applied to classes, methods, and variables	Can be applied to exception handling cases	Can be applied to objects
After declaring the final keyword, it can't be updated.	Whether an exception occurs or it does not, the finally keyword will run the crucial code.	The cleaning of objects during garbage collection is done using the finalize keyword.

This keyword is applied only when it is called.	It gets applied once the execution of the try-catch block is done.	It applies at the time of object cleaning.			

### • When to use the super keyword in Java?

The role of the super keyword in Java is to refer to the adjacent parent class object. It is generally used with variables, methods, and constructors. In addition to being a reference keyword for parent class objects, it can also be used to trigger the parent class methods and constructors.

### • What is a ClassLoader in Java?

A Java ClassLoader is used to load the classes in JRE in a dynamic manner. It is an important component in the Runtime Environment that loads the class into the memory part of the JRE.

It is because of ClassLoaders that the JRE doesn't have to have information about the files loaded to it.

### • What are the different types of ClassLoaders in Java?

There are three ClassLoader types in Java, as defined below:

#### • Bootstrap ClassLoader

Used for loading the important classes and internal classes of Java Development Kit (JDK). This ClassLoader runs only when it is called by the Java Virtual Machine (JVM).

#### • Extension ClassLoader

Used for loading classes from the extensions directory of the JDK. It is a child of the Bootstrap ClassLoader.

#### • System ClassLoader

Also called Application ClassLoader, it is used for loading the classes from the environment variable CLASSPATH. It is a child of the Extension ClassLoader.

- **Is it possible to access the members of a subclass if you create a superclass' object?**

No. It is not possible to access the subclass members. Only superclass members will be accessible.

- **How to define a functional interface in Java?**

We can use the `@FunctionalInterface` annotation in the Java 8 to define a functional interface.

- **How Lambda expressions and functional interface are interrelated?**

We can call the functional interface a large platform that comes with numerous expressions. The lambda expressions are one such part of this interface. This is the interrelation between the two.

- **What methods are used in Java 8 to define a number in functional interface?**

Generally, the static method and default method are used when a number is defined in a functional interface.

- **What are the things to know and guidelines related to functional interface in Java 8?**

Programmers need to follow these guidelines:

- Only a single method should be used to define the interface
- You can't define multiple abstracts
- Utilize `@FunctionalInterface` annotation in order to define a functional interface
- To define a number, you can use whatever method you want to
- In case you override the method of `java.lang.Object` class, it won't be counted as an abstract method.

- **What are the different types of functional interfaces in Java 8?**

These are the main types of functional interface:

- Consumer
  - Predicate
  - Supplier
  - Function (UnaryOperator and BinaryOperator)
- **State the biggest difference between Map and FlatMap.**

The primary difference between the two is that Map wraps the return value in the ordinal type, whereas FlatMap doesn't do it.

- **What is the primary benefit for which one should use Metaspace over PermGen?**

There is one big reason to go for Metaspace instead of PermGen. This reason is that the size of PermGen is fixed. As a result, it can't increase in a dynamic manner. On the other hand, the Metaspace does not have any limitations in terms of size. Its size can increase dynamically.

- **What is the difference between composition and aggregation in Java?**

Both composition and aggregation are associations in Java. The former is considered a strong association, while the latter is considered a weak association. Let's understand the differences between them with the below tabular comparison:

<b>Aggregation</b>	<b>Composition</b>
Weak	Strong
There is a relationship between classes	A class belongs to another class
Interrelated classes can be independent	Classes are dependent on each other.
As the classes can be independent, it is great for reusing the code	As the classes are not independent, code reusability becomes difficult

## **List of Java 8 Interview Questions**

**Here are some of the most common interview questions on Java 8:**

- What are the new features in Java SE 8?
- What are some of the main benefits of using Java 8?
- Define optional class.
- What is a functional interface in Java 8?
- Define MetaSpace.
- What is the meaning of the String::ValueOf expression?
- Explain the concept of streams in Java 8.
- What is Nashorn in Java 8?
- How is MetaSpace different from PermGen?
- What do you mean by method reference?
- Explain intermediate and terminal operations.
- Which are the most used terminal operations?
- Is it possible for a functional interface to inherit another interface?

- What is the difference between `findFirst()` and `findAny()`?
- Which are the key components of a Java stream?
- Which functional interfaces come pre-defined in Java 8?
- What does type interface mean?
- State the syntax of a lambda expression.
- What is the difference between collection and stream?
- Explain the role of JJS in Java 8.

## Java Interview FAQs

- **What is Java interview questions?**

The Java interview questions mean the concepts or things that are very likely to be asked to a candidate when he/she goes for the job interview. These are appropriate for candidates applying for jobs as:

- Java Developer
- Java Programmer
- Senior Java Developer
- Java Web Developer
- Java Android Developer
- Java EE Developer
- Java Engineer
- Java Technical Lead

- **Can you share some Java OOPS interview questions?**

Following OOPS concepts in Java interview questions can be asked to you:

- What is the meaning of OOPS in programming?
- What is the role of OOPS in Java?
- What are the primary features of OOPS?
- What are the pros and cons of OOPS?
- What is encapsulation?
- What is polymorphism?
- What is abstraction?
- Is Java a pure object-oriented programming language? If not, why?
- What is the difference between a class and an object?
- What are manipulators?
- What is the difference between constructor and method?
- What is a destructor?
- What are the different types of inheritance?
- What is the difference between OOP and procedural programming?
- Explain the difference between error and exception.
- Which are the most popular object-oriented programming languages?

- What is the difference between runtime polymorphism and compile-time polymorphism?
- Is it mandatory that objects will always be created from a class?
- What is a subclass?
- What is a superclass?
- What is static polymorphism?
- What is dynamic polymorphism?
- What is the difference between overriding and overloading?
- What is garbage collection?
- **What are the top Java spring boot interview questions?**

Here is the list of questions related to spring boot:

- What is spring boot in Java?
- Why use spring boot? What are its benefits?
- Which are the main components of spring boot?
- What is spring initializer?
- Explain the differences between @Controller and @RestController?
- What do you understand by dependency injection?
- How to define properties in spring boot?
- What do you mean by starter dependency?
- What is the role of @SpringBootApplication?
- Why do you use @ComponentScan?
- Explain start dependencies.
- Explain the differences between GetMapping and RequestMapping?
- What do you mean by Spring Boot CLI?
- Tell me about some of the most used CLI commands?
- **What are the most asked Java microservices interview questions?**

Here is the list of questions related to microservices that can be asked in your Java interview:

- What do you understand by microservices architecture?
- Which are the top microservices tools?
- Why do we use microservices?
- What is the role of reports and dashboards in microservices?
- Explain the difference between monolithic architecture and microservices?
- What is monolithic architecture?
- Tell about the key features of microservices.
- What is cohesion?
- What do you understand by coupling?
- What is the meaning of domain-driven design?
- What is bounded context?



- Explain which tests are used in microservices?
- Why is PACT used in microservices?
- **What are some Java interview questions for Selenium Tester?**

As a Selenium Automation Tester, you can expect the following interview questions on Java:

- What is data hiding in Java?
- Explain the concept of encapsulation?
- What do you understand by a tightly encapsulated class?
- Tell me about the getter and setter methods in Java?
- What does the Is-A relationship mean in Java?
- Explain the limitations of Java in Selenium testing?
- What is Java method overloading?
- Tell me about the same-origin policy and the way of handling it?
- How to implement inheritance in Java?
- Does Java support multiple inheritances using class? If not, why?
- Can you make use of super() and this() in a single constructor?
- **Which are the top Java concurrency interview questions?**

The interviewer may ask you the following concurrency questions in Java:

- What is concurrency?
- What do you mean by the executors framework?
- Explain the atomic operation?
- Explain the lifecycle of a thread?
- How to set up the environment for Java concurrency?
- What is an atomic operation in Java?
- Which atomic classes are used in the API of Java concurrency?
- What is an executor class?
- Explain the concept of lock interface in concurrency API?
- What is BlockingQueue in Java concurrency?
- What do you understand by FutureTask class?
- **What are the frequently asked Java collections interview questions?**

Here is the list of interview questions on the Java collections concept:

- What is Java collection?
- Explain the differences between collection and array in Java?
- Explain the difference between ArrayList and LinkedList?
- What is the difference between enumeration and iterator in Java?
- Tell me about the collection framework hierarchy?
- What is a priority queue?
- Explain the differences between HashSet and TreeSet in Java?

- State the differences between HashSet and HashMap?
- Is it possible to add a null element to HashSet?
- What are fail-fast and fail-safe iterators?
- Explain the differences between ListIterator and Iterator?
- **Which are the primary Java thread interview questions?**

Below is the list of common questions related to threading and multithreading in Java programming:

- Explain the concept of multithreading in Java?
- What benefits does multithreading offer?
- What are the different states of a thread lifecycle?
- How to create a Java thread?
- How to implement a thread in Java?
- Tell about the concept of thread priority.
- How do the threads in Java interact with each other?
- Explain ThreadLocal in Java.
- What are some ways to get thread safety?
- What is the reason behind a sleep() thread being static?
- What is a daemon thread, and how to create it?
- What is deadlock?
- Explain the concept of a thread pool and how can we create a thread pool.
- **Which are the top Java string interview questions?**

Following is the list of interview questions on Java string concept:

- Define the string in Java?
- What are the ways for string declaration in Java?
- What is the role of the string intern() method?
- Explain the differences between String and StringBuffer?
- When saving passwords in Java, developers choose a character array over a string. Why do they do so?
- What are the reasons behind string being immutable?
- State the differences between StringBuilder and StringBuffer in Java.
- How can you compare two strings in Java?
- Explain the role of the substring() method in Java?
- Can you check whether a string is empty or not? If yes, how?
- What are the ways for converting a string to a byte array?
- How to find the longest palindrome in a Java string?
- What is a string pool in Java?

- **What are the most asked Java inheritance interview questions?**

Recruiters or interviewers often ask these questions related to inheritance in Java:

- Explain Java inheritance.
- What is the purpose of using inheritance?
- Define the Is-A relationship in Java?
- How to implement inheritance in Java?
- How to create the subclass of a class?
- What are the main benefits of inheritance?
- State the difference between inheritance and multi-level inheritance?
- Explain hybrid inheritance.
- What are the different types of inheritance?
- Does Java support multi inheritance through class? If not, why?

- **What are the top Java 8 interview questions for 10 years experience?**

These tough Java interview questions can be asked to an experienced candidate:

- What is new in Java 8 compared to previous versions?
- What is a functional interface in Java 8?
- Explain the differences between a functional interface and a SAM interface.
- How to define a functional interface?
- What guidelines need to be followed for the functional interface?
- How to define a number in a functional interface?
- How are lambda expressions and functional interfaces interrelated?
- What are the main types of functional interfaces in Java 8?
- State the differences between Map and FlatMap.
- What are the benefits of using Metaspaces compared to PerGen?
- **What are the Java technical lead interview questions?**

In case you are applying for the role of Java Technical Lead, then expect some tricky and technical Java interview questions, as mentioned below:

- Can you debug a Java program while it's running? How?
- What is an asynchronous event?
- Which tools are best to use for testing Java code?
- Which tools are the best to probe Java memory leaks?
- Explain the decorator design pattern in Java.
- What are the key Java 8 features that can make the lives of programmers easier?
- What is the role of LDAP servers?
- Have you developed enterprise software or applications using Java? Share the name and your experience with it.

- What is an LDAP server and its uses?
- Explain the Spring MVC flow.
- What are RESTful web services in Java?
- What are some good ways to avoid a database deadlock?
- Explain the concepts of Spring security authentication and authorization.
- What is digest authentication in Spring security?
- What is SecurityContext in Spring security?
- State the use of AbstractSecurityInterceptor in spring security.
- What is a 2-way SSL, and why is it required?
- How to implement 2-way SSL using spring boot?
- Explain the design patterns in microservices architecture.

- **What are the most asked Java interview questions for 2 years experience?**

For developers or programmers working for a couple of years now, below are the top Java interview questions for 2 years experience:

- Is it possible to override a static method? If yes, how?
  - Which Java class is considered the base class?
  - State the difference between HashMap and HashSet.
  - Why Java strings are immutable?
  - What is a ClassPath in Java?
  - Explain the differences between StringBuffer and StringBuilder.
  - What is Java multithreading?
  - What is an applet in Java?
  - Explain garbage collection in Java.
  - What is inheritance in Java?
  - What is polymorphism in Java?
  - Explain Java serialization.
  - When should we use the transient variable?
  - Is it possible to call the start method twice? If yes, how?
  - How to make a class immutable in Java?
- 
- What is a Java copy constructor?
  - How the sorting of custom objects is done in Java?
  - Explain the marker interface in Java.
  - Explain the differences between LinkedList and ArrayList.