



# Advanced Java Interview Questions and Answers

## Introduction

More than 10 million global software developers are using Java, as it is the most widely used programming language in the world. Learning Java is beneficial for fresh graduates and final-year students to kick-start their careers in software development. Here in this blog, we have framed popular Java Interview Questions and Answers for the aspirants to face the interviews successfully. Explore our advanced Java course syllabus to get started.

## List of Advanced Java Interview Questions for Freshers

1. What is garbage collection in Java?
2. Define Lambda Expressions.

3. What is a thread in Java?
4. What is the syntax used to read and write data from the buffer?
5. Explain deadlock with its work on the program.
6. Define BufferedReader
7. What is Java NIO?
8. Explain JDBC drivers.
9. What is JavaMail?
10. What are the two packages used in JavaMail?

Enhance your development skills with our Advanced Java course in Chennai.

## **Advanced Java Interview Questions and Answers for Freshers**

### **1. What is garbage collection in Java?**

Garbage collection in Java is used to recognize and delete objects that are not needed by a program and are consuming resources unnecessarily. The garbage collection process is applied to free up space and reuse other objects. Java objects are subject to garbage collection when they are inaccessible to the Java program where they are initiated.

### **2. Define Lambda Expressions.**

Lambda Expressions are a new feature in Java 8, and they provide a simpler way to work with interfaces that have only a method used in places where the programmer uses anonymous classes. If the programmer is required to write some lines to complete what he wants to do with the use of a print statement, then he can write them in a single line using a lambda expression. The usage of lambda expressions increases the program's efficiency by reducing the lines of code.

### **3. What is a thread in Java?**

The thread is a single continuous flow of control within the process, where every process can consist of two or more "threads". It is created and maintained by the `java.lang.Thread` class. Each process or application in Java programming language includes at least one thread that depends on the requirements of the programmer. The process includes multiple threads to manage various tasks like memory management and I/O.

## 4. What is the syntax used to read and write data from the buffer?

Syntax for reading data from a Buffer

```
int dataobject = inChannel.read(buff);
```

Syntax for writing data from a Buffer

```
int dataobject = inChannel.write(buff);
```

## 5. Explain deadlock with its work on the program.

The deadlock occurs when two or more threads are blocked on the lock that is holding a lock as per the thread. If thread 1 is holding lock 1 and waiting to obtain lock 2, but thread 2 is holding lock 2 and waiting to obtain lock 1, it creates a situation where all threads holding the locks are blocked, for it will never release the locks holding, and none of the waiting threads will ever run.

## 6. Define BufferedReader.

BufferedReader is used to read text from the input stream and buffers the characters into a character array. Reading chunks of data from a stream is more effective than reading characters at a time. BufferedReader is more efficient and faster for streamlining the size of data.

**Take your Knowledge  
Test Report**

**Check your Score**



## 7. What is Java NIO?

A new I/O package is included in the Java SDK in Java 1.4, and it is known as Java.Nio. It is the expansion to Java I/O as the classes in the package implement inputs and outputs in a non-blocking way. The package Java.Nio is used to fix problems that developers could execute using Java.io classes to work with the file system.

## 8. Explain JDBC drivers.

JDBC API will be implemented in the driver using a Java library that contains classes. As all JDBC drivers implement the same interfaces, it's not difficult to change the data source for application users. For instance, if an application uses an SQLite database, then the developer should decide that he wants to use a MySQL database, and all we have to do is use the MySQL JDBC driver for SQLite. JDBC contains two packages: `java.sql` as the core JDBC and `javax.sql` for optional JDBC. The APIs in the `java.sql` package are needed when working with database servers.

## 9. What is JavaMail?

JavaMail is a widely used API (Application Programming Interface) for facilitating mailing operations such as composing, reading, sending, and receiving electronic mail on the internet. It is a protocol-independent framework available with Java SE (Standard Edition) and associated with the Java EE (Enterprise Edition) platform. The JavaMail API will be used for various events, and it can be used to register the user for sending notifications, set and update the new password for forgetting the password, and send update notifications.

## 10. What are the two packages used in JavaMail?

In JavaMail, there are two packages used that contain various classes. The first one is `javax.mail` and the second one is `javax.mail.activation`.

Enrich your software development skills with our advanced Java tutorial for beginners.

## List of Advanced Java Interview Questions for Experienced

1. Why should we use multiple threads in the application, and why not with the main thread?
2. What are the advantages and disadvantages of Java Sockets?
3. What is synchronization, and how does it benefit multithreading?
4. Explain the use of the JSTL core tag with an example.
5. Define session tracking and explain the common methods of session tracking.
6. Can JSP allow the user of exception implicit objects in pages?
7. Can a request be forwarded from a JSP page to a servlet page?
8. Is JSP an extensible technology?
9. Explain Static Initializer.

10. What is the difference between a transaction and a distributed transaction?

Know the best advanced Java salary for freshers and experienced professionals.

## Advanced Java Technical Interview Questions and Answers for Experienced

### 1. Why should we use multiple threads in the application, and why not with the main thread?

There are two main reasons to explain

- If we want to perform a task that takes a long time, it might be queried from the database, or it should be fetched from the internet. It can be done easily with the main thread, but the code within every main thread executes linearly. The main thread will not be able to perform anything else as it is waiting for the data.
- When the execution of the main thread is suspended, it has to wait for the data to be returned before it will execute the next line of code. This will appear as if our application has died or is frozen to the user.

We can create multiple threads to execute the long-running task on those threads, which helps free up the space using the main thread. It is continually executed, and this process is called multithreading. It can report progress or accept user input during the long-running process that continues to execute in the background.

### 2. What are the advantages and disadvantages of Java Sockets?

- **Advantages of Java Sockets:** Sockets in Java programming are used for their flexibility and easy communication protocols. It leads to low network traffic, like HTML forms and CGI scripts that generate and transfer the complete web pages for each new data request.
- **Limitations of Java Sockets:** The communication that occurs through sockets enables the sending of packages of raw data between applications.

### 3. What is synchronization, and how does it benefit multithreading?

Synchronization is the process of managing the access of multiple threads to shared resources in multithreading. We can synchronize methods and statements in Java, and whenever we synchronize methods in a Java program.

It enables only one thread to execute at a time. If a thread is using a synchronized method, all the other threads that want to run the same method or any other synchronized method initiated in the same class will suspend until the execution of the executing thread is completed successfully.

**Take your Knowledge  
Test Report**

**Check your Score**



### 4. Explain the use of the JSTL core tag with an example.

The JSTL `<c:forEach>` is an iteration tag used to loop or repeat the nested body content for a particular number of times. It is an alternative to Java loops such as while, do-while, or for.

#### Example

```
<%@ taglib url = http://java.sun.com/jsp/jstl/core prefix = "c" %>
```

```
<html> <head>
```

```
<title> forEach Core Tag </title>
```

```
</head>
```

```
<body>
```

```
<c:forEach var = "j" begin = "1" end = "4">
```

```
Item <c:out value = "${j}" /><p>
```

```
</c:forEach>
```

```
</body>
```

```
</html>
```

### **Output:**

Item 1

Item 2

Item 3

Item 4

## **5. Define session tracking and explain the common methods of session tracking.**

A session means a specific interval of time used to track and maintain user data for a particular time frame. Session tracking is a mechanism used by the web container to manage the state of a user. The HTTP and web servers are stateless protocols. All requests and responses are unique, and every time a user sends requests to the server, it will be treated as a new request. By that time, we can use session tracking to prevent and maintain the state of a user. The following are the common methods used in session tracking

- Cookies
- HttpSession
- Hidden Form Field
- URL Rewriting

## **6. Can JSP allow the user of exception implicit objects in pages?**

The exception implicit object will be created for the JSP error page, and it is defined using the `ErrorPage` attribute. It can't be used in the implicit exception object on any JSP page.

## 7. Can a request be forwarded from a JSP page to a servlet page?

Yes, a request can be forwarded from the JSP page to the Servlet using the JSP “forward action” tag and by mentioning the URL pattern of the Servlet.

## 8. Is JSP an extensible technology?

JSP technology is extensible by the development of tags or custom actions that are encapsulated in tag libraries.

## 9. Explain Static Initializer.

The static initializer will be defined as a static `{}` block of code that executes only once before the constructor or main method is called for action. It is used when the user wants to run a complicated calculation to define an 'n' value or if its value is fetched from a database.

### Example:

***StaticInitializerExample.Java***

```
class StaticInitializerExample  
  
    public static int n;  
  
    static  
  
    {  
  
        n = 12;  
  
    }  
  
}
```

## 10. What is the difference between a transaction and a distributed transaction?

A transaction is defined as a logical unit of work that either succeeds or creates no result, while a distributed transaction can access and update data on multiple computing networked resources.

## Conclusion

Advanced Java is the most important skill as it contains Java Enterprise Edition, Servlet Programming, Web Services, The Persistence API, etc that are used for web and enterprise application development that follow client-server architecture. Increase your profile visibility with our software training institute in Chennai.