


Top 75 Java Interview Questions You Must Prepare In 2019

677.3K Views



Aayushi Johari
[Published on Jan 02, 2019](#)

 Bookmark

Java Interview Questions

In this Java Interview Questions blog, I am going to list some of the most important Java Interview Questions and Answers which will set you apart in the interview process. Java is used by approx 10 Million developers worldwide to develop applications for 15 Billion devices supporting Java. It is also used to create applications for trending technologies like Big Data to household devices like Mobiles and DTH boxes. And hence today, **Java is used everywhere!** This is the reason why [Java Certification](#) is most in-demand certification in programming domain.

We have compiled a list of top Java interview questions which are classified into 7 sections, namely:

1. [Basic Interview Questions](#)
2. [OOps Interview questions](#)
3. [JDBC Interview questions](#)
4. [Spring Interview questions](#)
5. [Hibernate Interview questions](#)
6. [JSP Interview questions](#)
7. [Exception and thread Interview questions](#)

Java Interview Questions and Answers | Edureka

Java Interview Questions and Answers | Java Tutorial | Java Online Tr...



As a Java professional, it is essential to know the right buzzwords, learn the right technologies and prepare the right answers to commonly asked Java Interview Questions. Here's a definitive list of top Java Interview Questions that will guarantee a breeze-through to the next level.

In case you attended any Java interview recently, or have additional questions beyond what we covered, we encourage you to post them in our [QnA Forum](#). Our expert team will get back to you at the earliest.


So let's get started with the first set of basic Java Interview Questions.

Basic Java Interview Questions


Q1. Explain JDK, JRE and JVM?

JDK	JRE	JVM
It stands for Java Development Kit.	It stands for Java Runtime Environment.	It stands for Java Virtual Machine.

Subscribe to our Newsletter, and get personalized recommendations.



Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)

Along with JRE, it includes an interpreter/loader, a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development. In short, it contains JRE + development tools.	It implements the JVM (Java Virtual Machine) 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 Java program. Basically, it's an implementation of the JVM which physically exists.	JVM follows three notations: Specification(document that describes the implementation of the Java virtual machine), Implementation (program that meets the requirements of JVM specification) and Runtime Instance (instance of JVM is created whenever you write a java command on the command prompt and run class).
--	--	--

JDK vs JRE vs JVM

Q2. Explain public static void main(String args[]).

- **public** : Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.
- **static** : It is a keyword in java which identifies it is class based i.e it can be accessed without creating the instance of a Class.
- **void** : It is the return type of the method. Void defines the method which will not return any value.
- **main**: It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.
- **String args[]** : It is the parameter passed to the main method.

Q3. Why Java is platform independent?

Platform independent practically means “write once run anywhere”. Java is called so because of its byte codes which can run on any system irrespective of its underlying operating system.

Q4. Why java is not 100% Object-oriented?


Java is not 100% Object-oriented because it makes use of eight primitive datatypes such as boolean, byte, char, int, float, double, long, short which are **not objects**.

Q5. What are wrapper classes?

Wrapper classes converts the java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they “wrap” the primitive data type into an object of that class. Refer to the below image which displays different primitive type, wrapper class and constructor argument.

Primitive	Wrapper Class	Constructor Argument
boolean	Boolean	boolean or String
byte	Byte	byte or String
char	Character	char
int	Integer	int or String
float	Float	float, double or String
double	Double	double or String
long	Long	long or String

Subscribe to our Newsletter, and get personalized recommendations.

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

- 1. Default constructor
- 2. Parameterized constructor

Q7. What is singleton class and how can we make a class singleton?

Singleton class is a class whose only one instance can be created at any given time, in one JVM. A class can be made singleton by making its constructor private.

Q8. What is the difference between Array list and vector?

Array List	Vector
Array List is not synchronized.	Vector is synchronized.
Array List is fast as it's non-synchronized.	Vector is slow as it is thread safe.
If an element is inserted into the Array List, it increases its Array size by 50%.	Vector defaults to doubling size of its array.
Array List does not define the increment size.	Vector defines the increment size.
Array List can only use Iterator for traversing an Array List.	Except Hashtable, Vector is the only other class which uses both Enumeration and Iterator.

Q9. What is the difference between equals() and == ?

Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic. “==” or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. *public boolean equals(Object o)* is the method provided by the Object class. The default implementation uses == operator to compare two objects. For example: method can be overridden like String class. equals() method is used to compare the values of two objects.

Q10. What are the differences between Heap and Stack Memory?

The major difference between Heap and Stack memory are:

Features	Stack	Heap
Memory	Stack memory is used only by one thread of execution.	Heap memory is used by all the parts of the application.
Access	Stack memory can't be accessed by other threads.	Objects stored in the heap are globally accessible.
Memory Management	Follows LIFO manner to free memory.	Memory management is based on generation associated to each object.
Lifetime	Exists until the end of execution of the thread.	Heap memory lives from the start till the end of application execution.
Usage	Stack memory only contains local primitive and reference variables to objects in heap space.	Whenever an object is created, it's always stored in the Heap space.


In case you are facing any challenges with these java interview questions, please comment on your problems in the section below.

OOPS Java Interview Questions:

Q1. What is Polymorphism?



Subscribe to our Newsletter, and get personalized recommendations. ✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

Polymorphism is briefly described as “one interface, many implementations”. Polymorphism is a characteristic of being able to assign a different meaning or usage to something in different contexts – specifically, to allow an entity such as a variable, a function, or an object to have more than one form. There are two types of polymorphism:

- 1. Compile time polymorphism
- 2. Run time polymorphism

Compile time polymorphism is method overloading whereas Runtime time polymorphism is done using inheritance and interface.

Q2. What is runtime polymorphism or dynamic method dispatch?

In Java, runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time. In this process, an overridden method is called through the reference variable of a superclass. Let’s take a look at the example below to understand it better.

```
1  class Car {
2  void run()
3  {
4  System.out.println("car is running");
5  }
6  }
7  class Audi extends Car {
8  void run()
9  {
10 System.out.println("Audi is running safely with 100km");
11 }
12 public static void main(String args[])
13 {
14 Car b= new Audi();    //upcasting
15 b.run();
16 }
17 }
```

Q3. What is the difference between abstract classes and interfaces?

Abstract Class	Interfaces
An abstract class can provide complete, default code and/or just the details that have to be overridden.	An interface cannot provide any code at all,just the signature.
In case of abstract class, a class may extend only one abstract class.	A Class may implement several interfaces.
An abstract class can have non-abstract methods.	All methods of an Interface are abstract.
An abstract class can have instance variables.	An Interface cannot have instance variables
An abstract class can have any visibility: public, private, protected.	An Interface visibility must be public (or) none.
If we add a new method to an abstract class then we have the option of providing default implementation and therefore all the existing code might work properly	If we add a new method to an Interface then we have to track down all the implementations of the interface and define implementation for the new method
An abstract class can contain constructors	An Interface cannot contain constructors
Abstract classes are fast	Interfaces are slow as it requires extra indirection to find corresponding method in the actual class


Q4. What is method overloading and method overriding?

Method Overloading :

- In Method Overloading, Methods of the same class shares the same name but each method must have different number of parameters or parameters having different types and order.
- Method Overloading is to “add” or “extend” more to method’s behavior

Subscribe to our Newsletter, and get personalized recommendations.

✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

```

1  class Adder {
2  static int add(int a, int b)
3  {
4  return a+b;
5  }
6  static double add( double a, double b)
7  {
8  return a+b;
9  }
10 public static void main(String args[])
11 {
12 System.out.println(Adder.add(11,11));
13 System.out.println(Adder.add(12.3,12.6));
14 }

```

Method Overriding:

- In Method Overriding, sub class have the same method with same name and exactly the same number and type of parameters and same return type as a super class.
- Method Overriding is to “Change” existing behavior of method.
- It is a run time polymorphism.
- The methods must have same signature.
- It always requires inheritance in Method Overriding.

Let's take a look at the example below to understand it better.

```

1  class Car {
2  void run(){
3  System.out.println("car is running");
4  }
5  Class Audi extends Car{
6  void run()
7  {
8  System.out.println("Audi is running safely with 100km");
9  }
10 public static void main( String args[])
11 {
12 Car b=new Audi();
13 b.run();
14 }
15 }

```

Q5. Can you override a private or static method in Java?

You cannot override a private or static method in Java. If you create a similar method with same return type and same method arguments in child class then it will hide the super class method; this is known as method hiding. Similarly, you cannot override a private method in sub class because it's not accessible there. What you can do is create another private method with the same name in the child class. Let's take a look at the example below to understand it better.

```

1  class Base {
2  private static void display() {
3  System.out.println("Static or class method from Base");
4  }
5  public void print() {
6  System.out.println("Non-static or instance method from Base");
7  }
8  class Derived extends Base {
9  private static void display() {
10 System.out.println("Static or class method from Derived");
11 }
12 public void print() {
13 System.out.println("Non-static or instance method from Derived");
14 }
15 public class test {
16 public static void main(String args[])
17 {
18 Base obj= new Derived();
19 obj1.display();
20 obj1.print();
21 }

```

Subscribe to our Newsletter, and get personalized recommendations. ✕

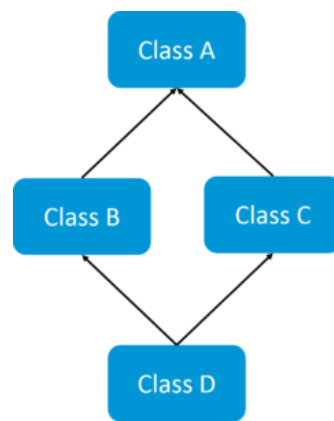


Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)



If a child class inherits the property from multiple classes is known as multiple inheritance. Java does not allow to extend multiple classes.

The problem with multiple inheritance is that if multiple parent classes have a same method name, then at runtime it becomes difficult for the compiler to decide which method to execute from the child class.

Therefore, Java doesn't support multiple inheritance. The problem is commonly referred as Diamond Problem.

Q7. What is association?

Association is a relationship where all object have their own lifecycle and there is no owner. Let's take an example of Teacher and Student. Multiple students can associate with a single teacher and a single student can associate with multiple teachers but there is no ownership between the objects and both have their own lifecycle. These relationship can be one to one, One to many, many to one and many to many.

Q8. What do you mean by aggregation?

Aggregation is a specialized form of Association where all object have their own lifecycle but there is ownership and child object can not belongs to another parent object. Let's take an example of Department and teacher. A single teacher can not belongs to multiple departments, but if we delete the department teacher object will not destroy.

Q9. What is composition in Java?

Composition is again specialized form of Aggregation and we can call this as a "death" relationship. It is a strong type of Aggregation. Child object dose not have their lifecycle and if parent object deletes all child object will also be deleted. Let's take again an example of relationship between House and rooms. House can contain multiple rooms there is no independent life of room and any room can not belongs to two different house if we delete the house room will automatically delete.

In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka!

Servlets Interview Questions

Q1. What is a servlet?

- Java Servlet is server side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.
- The javax.servlet and javax.servlet.http packages provide interfaces and classes for writing our own servlets.
- All servlets must implement the javax.servlet.Servlet interface, which defines servlet lifecycle methods. When implementing a generic service, we can extend the GenericServlet class provided with the Java Servlet API. The HttpServlet class provides methods, such as doGet() and doPost(), for handling HTTP-specific services.
- Most of the times, web applications are accessed using HTTP protocol and thats why we mostly extend HttpServlet class. Servlet API hierarchy is shown in below image.

Subscribe to our Newsletter, and get personalized recommendations. ✕

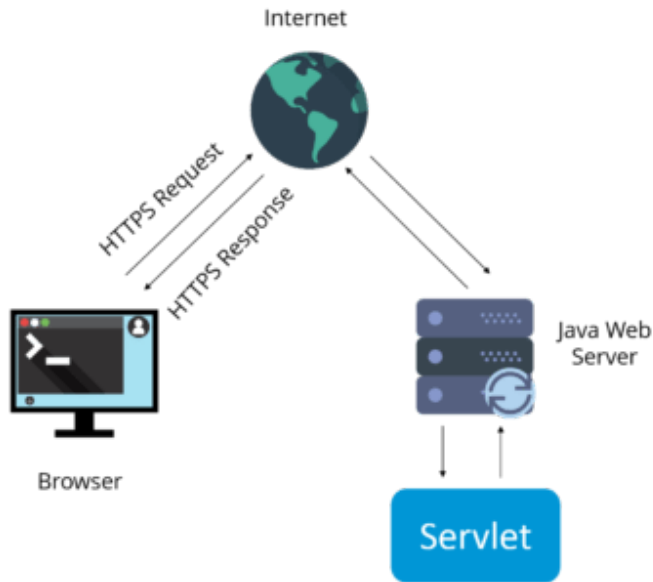


Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)



Q2. What are the differences between Get and Post methods?

Get	Post
Limited amount of data can be sent because data is sent in header.	Large amount of data can be sent because data is sent in body.
Not Secured because data is exposed in URL bar.	Secured because data is not exposed in URL bar.
Can be bookmarked	Cannot be bookmarked
Idempotent	Non-Idempotent
It is more efficient and used than Post	It is less efficient and used

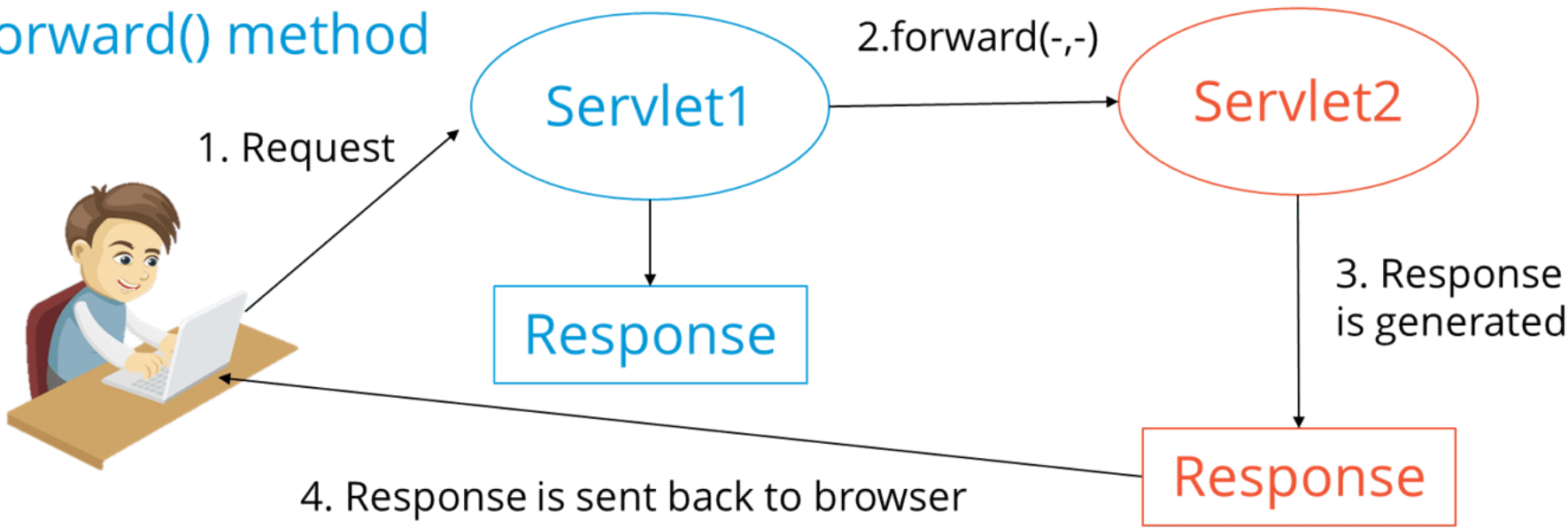
Q3. What is Request Dispatcher?

RequestDispatcher interface is used to forward the request to another resource that can be HTML, JSP or another servlet in same application. We can also use this to include the content of another resource to the response.

There are two methods defined in this interface:

- 1.void forward()
- 2.void include()


forward() method



include() method



Subscribe to our Newsletter, and get personalized recommendations. ✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

Q4. What are the differences between forward() method and sendRedirect() methods?

Forward() method	SendRedirect() method
forward() sends the same request to another resource.	sendRedirect() method sends new request always because it uses the URL bar of the browser.
forward() method works at server side.	sendRedirect() method works at client side.
forward() method works within the server only.	sendRedirect() method works within and outside the server.

Q5. What is the life-cycle of a servlet?

There are 5 stages in the lifecycle of a servlet:



Java, J2EE & SOA Certification Training

- [Instructor-led Sessions](#)
- [Real-life Case Studies](#)
- [Assignments](#)
- [Lifetime Access](#)

[Explore Curriculum](#)

1. Servlet is loaded
2. Servlet is instantiated
3. Servlet is initialized
4. Service the request
5. Servlet is destroyed


Q6. How does cookies work in Servlets?


- Cookies are text data sent by server to the client and it gets saved at the client local machine.
- Servlet API provides cookies support through javax.servlet.http.Cookie class that implements Serializable and Cloneable interfaces.
- HttpServletRequest getCookies() method is provided to get the array of Cookies from request, since there is no point of adding Cookie to request, there are no methods to set or add cookie to request.
- Similarly HttpServletResponse addCookie(Cookie c) method is provided to attach cookie in response header, there are no getter methods for cookie.

Q7. What are the differences between ServletContext vs ServletConfig?

The difference between ServletContext and ServletConfig in Servlets JSP is in below tabular format

Subscribe to our Newsletter, and get personalized recommendations. ✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

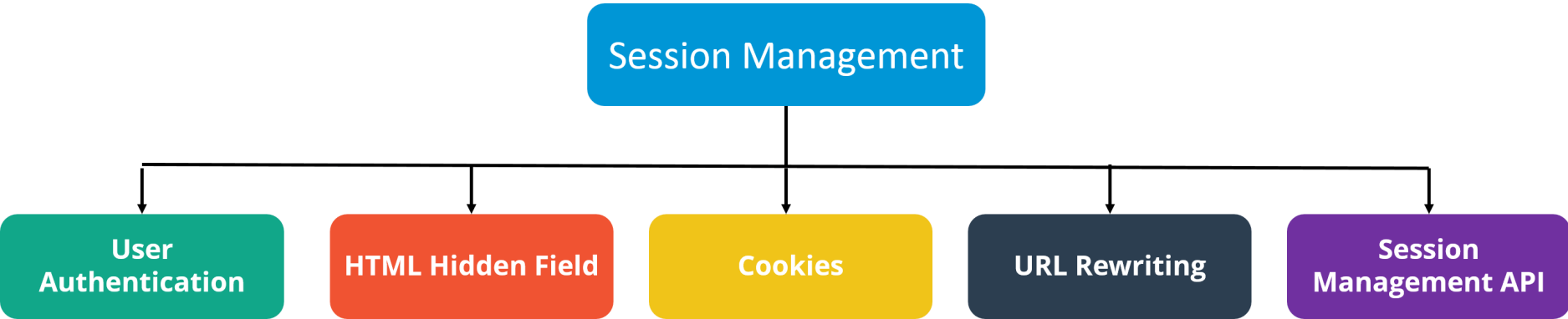
getServletConfig() method is used to get the config object	getServletContext() method is used to get the context object.
for example shopping cart of a user is a specific to particular user so here we can use servlet config	To get the MIME type of a file or application session related information is stored using servlet context object.

Q8. What are the different methods of session management in servlets?

Session is a conversational state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.

Some of the common ways of session management in servlets are:

- 1. User Authentication
- 2. HTML Hidden Field
- 3. Cookies
- 4. URL Rewriting
- 5. Session Management API



In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

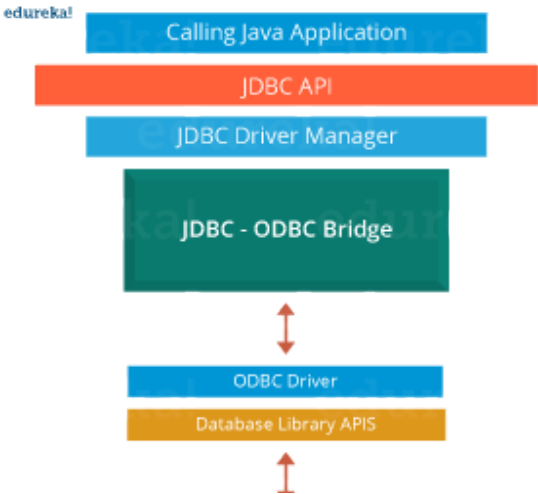
Get Java Certified

JDBC Interview Questions

Q1. What is JDBC Driver?

JDBC Driver is a software component that enables java application to interact with the database. There are 4 types of JDBC drivers:

- 1. JDBC-ODBC bridge driver
- 2. Native-API driver (partially java driver)
- 3. Network Protocol driver (fully java driver)
- 4. Thin driver (fully java driver)



Subscribe to our Newsletter, and get personalized recommendations.

Sign up with Google

Signup with Facebook

Already have an account? [Sign in.](#)

- Closing connection

Q3. What are the JDBC API components?

The java.sql package contains interfaces and classes for JDBC API.

Interfaces:

- Connection
- Statement
- PreparedStatement
- ResultSet
- ResultSetMetaData
- DatabaseMetaData
- CallableStatement etc.

Classes:

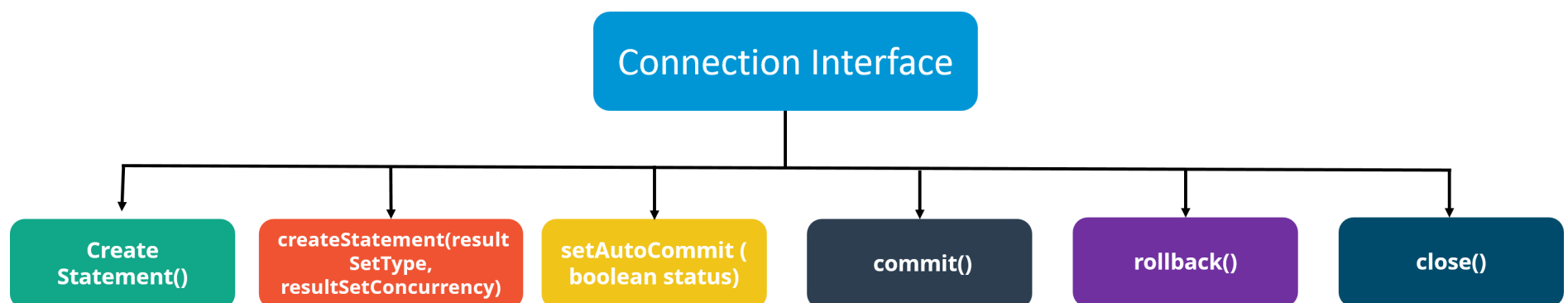
- DriverManager
- Blob
- Clob
- Types
- SQLException etc.

Q4. What is the role of JDBC DriverManager class?

The DriverManager class manages the registered drivers. It can be used to register and unregister drivers. It provides factory method that returns the instance of Connection.

Q5. What is JDBC Connection interface?

The Connection interface maintains a session with the database. It can be used for transaction management. It provides factory methods that returns the instance of Statement, PreparedStatement, CallableStatement and DatabaseMetaData.



Q6. What is the purpose of JDBC ResultSet interface?

The ResultSet object represents a row of a table. It can be used to change the cursor pointer and get the information from the database.

Q7. What is JDBC ResultSetMetaData interface?

The ResultSetMetaData interface returns the information of table such as total number of columns, column name, column type etc.

Q8. What is JDBC DatabaseMetaData interface?

The DatabaseMetaData interface returns the information of the database such as username, driver name, driver version, number of tables, number of views etc.

Q9. What do you mean by batch processing in JDBC?

Batch processing helps you to group related SQL statements into a batch and execute them instead of executing a single query. By using batch processing technique in JDBC, you can execute multiple queries which makes the performance faster.

Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)

Statement ***executeQuery(String query)*** is used to execute Select queries and returns the ResultSet. ResultSet returned is never null even if there are no records matching the query. When executing select queries we should use executeQuery method so that if someone tries to execute insert/update statement it will throw java.sql.SQLException with message “executeQuery method can not be used for update”.

Statement ***executeUpdate(String query)*** is used to execute Insert/Update/Delete (DML) statements or DDL statements that returns nothing. The output is int and equals to the row count for SQL Data Manipulation Language (DML) statements. For DDL statements, the output is 0.

You should use execute() method only when you are not sure about the type of statement else use executeQuery or executeUpdate method.

In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka!

Spring Interview Questions

Q1. What is a Spring?

Wikipedia defines the Spring framework as “an application framework and inversion of control container for the Java platform. The framework’s core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.” Spring is essentially a lightweight, integrated framework that can be used for developing enterprise applications in java.

Q2. Name the different modules of the Spring framework.

Some of the important Spring Framework modules are:

- Spring Context – for dependency injection.
- Spring AOP – for aspect oriented programming.
- Spring DAO – for database operations using DAO pattern
- Spring JDBC – for JDBC and DataSource support.
- Spring ORM – for ORM tools support such as Hibernate
- Spring Web Module – for creating web applications.
- Spring MVC – Model-View-Controller implementation for creating web applications, web services etc.

Subscribe to our Newsletter, and get personalized recommendations. 



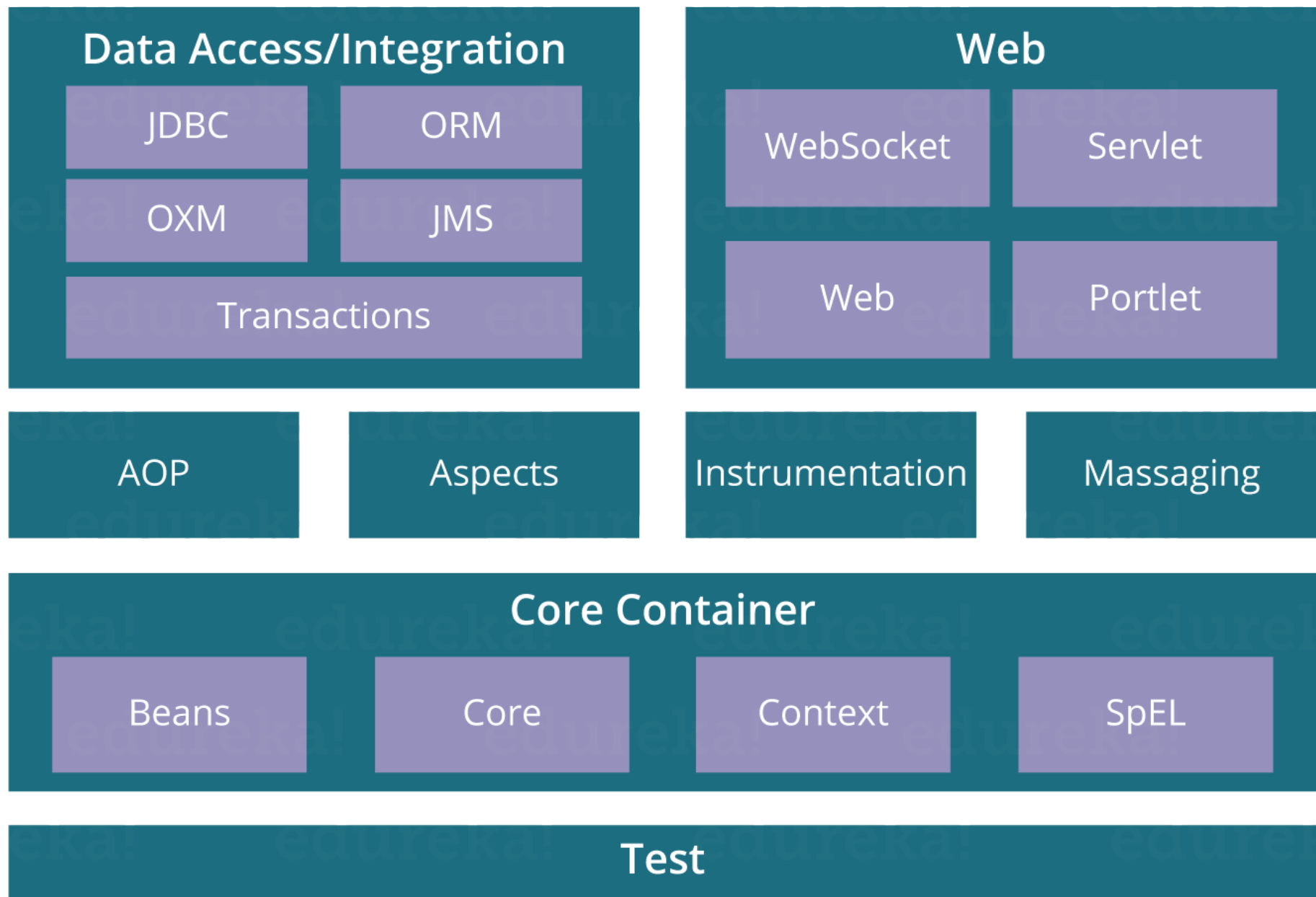
Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)

edureka!



Q3. List some of the important annotations in annotation-based Spring configuration.

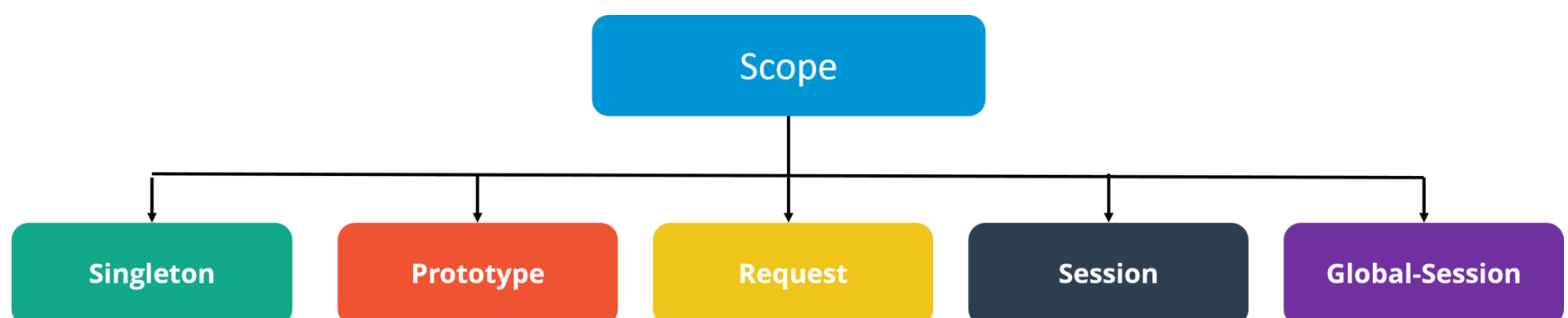
The important annotations are:

- @Required
- @Autowired
- @Qualifier
- @Resource
- @PostConstruct
- @PreDestroy

Q4. Explain Bean in Spring and List the different Scopes of Spring bean.

Beans are objects that form the backbone of a Spring application. They are managed by the Spring IoC container. In other words, a bean is an object that is instantiated, assembled, and managed by a Spring IoC container.

There are five Scopes defined in Spring beans.



Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google

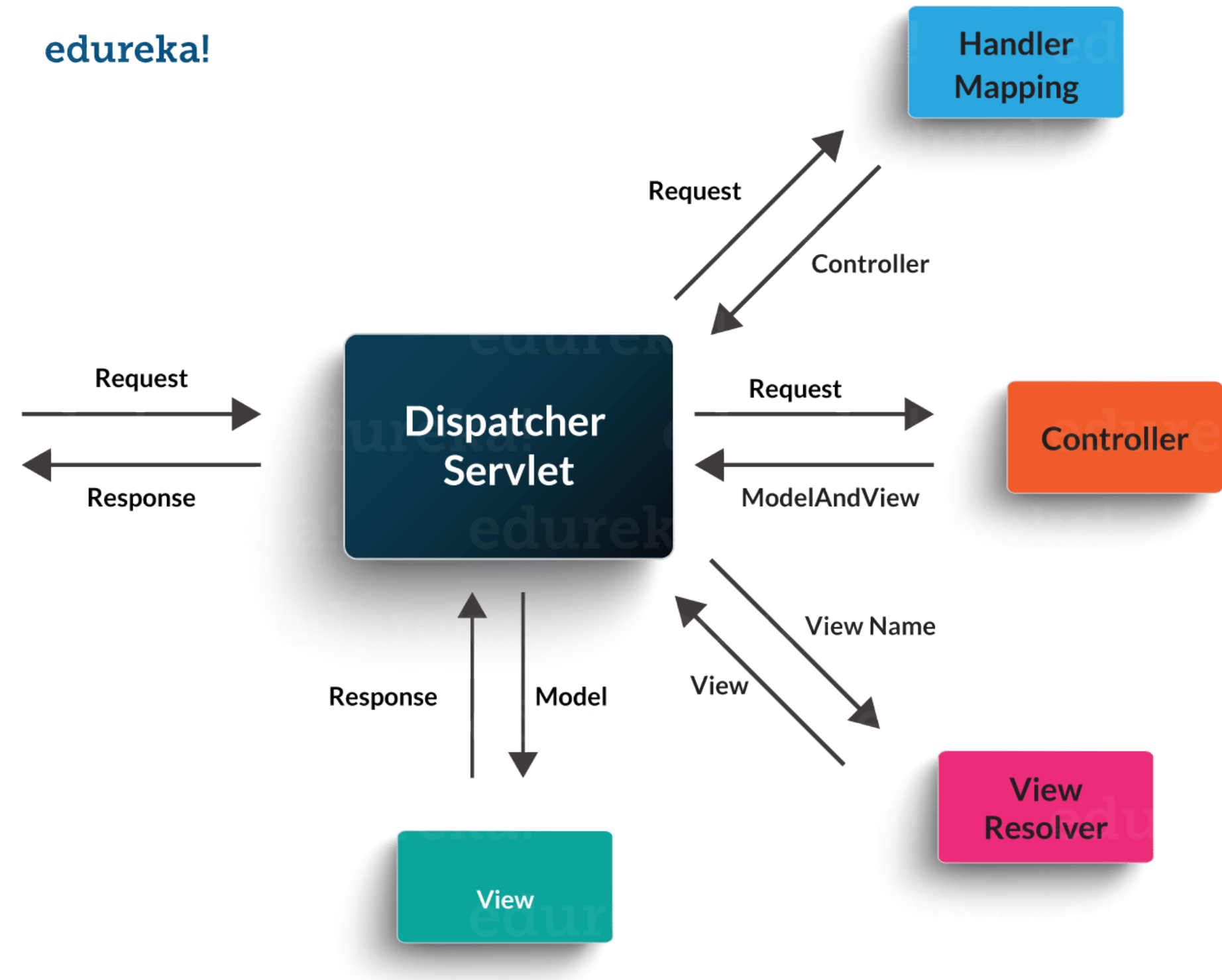


Signup with Facebook

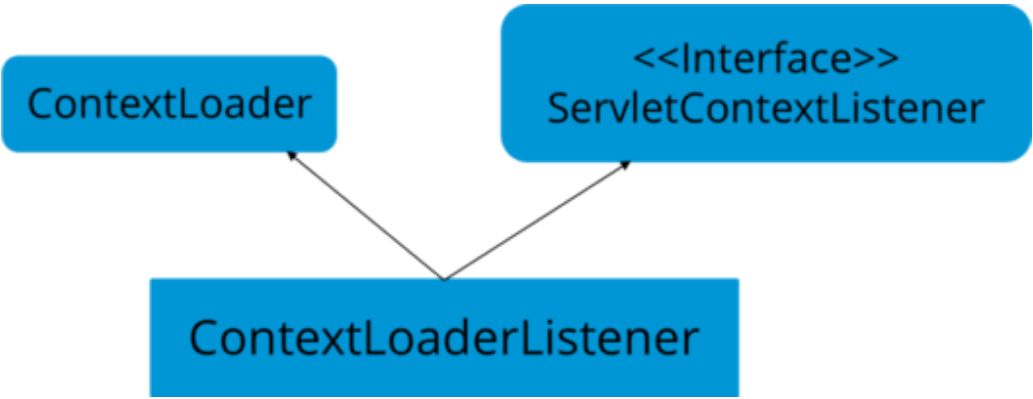
Already have an account? [Sign in.](#)

Q5. Explain the role of DispatcherServlet and ContextLoaderListener.

DispatcherServlet is basically the front controller in the Spring MVC application as it loads the spring bean configuration file and initializes all the beans that have been configured. If annotations are enabled, it also scans the packages to configure any bean annotated with @Component, @Controller, @Repository or @Service annotations.



ContextLoaderListener, on the other hand, is the listener to start up and shut down the WebApplicationContext in Spring root. Some of its important functions includes tying up the lifecycle of Application Context to the lifecycle of the ServletContext and automating the creation of ApplicationContext.




Q6. What are the differences between constructor injection and setter injection?

No.	Constructor Injection	Setter Injection
-----	-----------------------	------------------

Subscribe to our Newsletter, and get personalized recommendations.

✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

Autowiring enables the programmer to inject the bean automatically. We don't need to write explicit injection logic. Let's see the code to inject bean using dependency injection.

```
1. <bean id="emp" class="com.javatpoint.Employee" autowire="byName" />
```

The autowiring modes are given below:

No.	Mode	Description
1)	no	this is the default mode, it means autowiring is not enabled.
2)	byName	Injects the bean based on the property name. It uses setter method.
3)	byType	Injects the bean based on the property type. It uses setter method.
4)	constructor	It injects the bean using constructor

Q8. How to handle exceptions in Spring MVC Framework?

Spring MVC Framework provides the following ways to help us achieving robust exception handling.

Controller Based:

We can define exception handler methods in our controller classes. All we need is to annotate these methods with `@ExceptionHandler` annotation.

Global Exception Handler:

Exception Handling is a cross-cutting concern and Spring provides `@ControllerAdvice` annotation that we can use with any class to define our global exception handler.

HandlerExceptionHandler implementation:

For generic exceptions, most of the times we serve static pages. Spring Framework provides `HandlerExceptionHandler` interface that we can implement to create global exception handler. The reason behind this additional way to define global exception handler is that Spring framework also provides default implementation classes that we can define in our spring bean configuration file to get spring framework exception handling benefits.

Q9. What are some of the important Spring annotations which you have used?

Some of the Spring annotations that I have used in my project are:

@Controller – for controller classes in Spring MVC project.

@RequestMapping – for configuring URI mapping in controller handler methods. This is a very important annotation, so you should go through Spring MVC RequestMapping Annotation Examples

@ResponseBody – for sending Object as response, usually for sending XML or JSON data as response.



[Java,J2EE & SOA Certification Training](#)

Watch The Course Preview


@PathVariable – for mapping dynamic values from the URI to handler method arguments.

@Autowired – for autowiring dependencies in spring beans.

@Qualifier – with `@Autowired` annotation to avoid confusion when multiple instances of bean type is present.

@Service – for service classes.

Subscribe to our Newsletter, and get personalized recommendations.

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

We can use Spring ORM module to integrate Spring and Hibernate frameworks, if you are using Hibernate 3+ where SessionFactory provides current session, then you should avoid using HibernateTemplate or HibernateDaoSupport classes and better to use DAO pattern with dependency injection for the integration.

Also Spring ORM provides support for using Spring declarative transaction management, so you should utilize that rather than going for hibernate boiler-plate code for transaction management.

In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka!

Hibernate Interview Questions

Q1. What is Hibernate Framework?

Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables. Hibernate is java based ORM tool that provides framework for mapping application domain objects to the relational database tables and vice versa.

Hibernate provides reference implementation of Java Persistence API, that makes it a great choice as ORM tool with benefits of loose coupling. We can use Hibernate persistence API for CRUD operations. Hibernate framework provide option to map plain old java objects to traditional database tables with the use of JPA annotations as well as XML based configuration.

Similarly hibernate configurations are flexible and can be done from XML configuration file as well as programmatically.

Q2. What are the important benefits of using Hibernate Framework?

Some of the important benefits of using hibernate framework are:

1. Hibernate eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources, so we can focus on business logic.
2. Hibernate framework provides support for XML as well as JPA annotations, that makes our code implementation independent.
3. Hibernate provides a powerful query language (HQL) that is similar to SQL. However, HQL is fully object-oriented and understands concepts like inheritance, polymorphism and association.
4. Hibernate is an open source project from Red Hat Community and used worldwide. This makes it a better choice than others because learning curve is small and there are tons of online documentations and help is easily available in forums.
5. Hibernate is easy to integrate with other Java EE frameworks, it's so popular that Spring Framework provides built-in support for integrating hibernate with Spring applications.
6. Hibernate supports lazy initialization using proxy objects and perform actual database queries only when it's required.
7. Hibernate cache helps us in getting better performance.
8. For database vendor specific feature, hibernate is suitable because we can also execute native sql queries.

Overall hibernate is the best choice in current market for ORM tool, it contains all the features that you will ever need in an ORM tool.

Q3. Explain Hibernate architecture?

Subscribe to our Newsletter, and get personalized recommendations. ✕

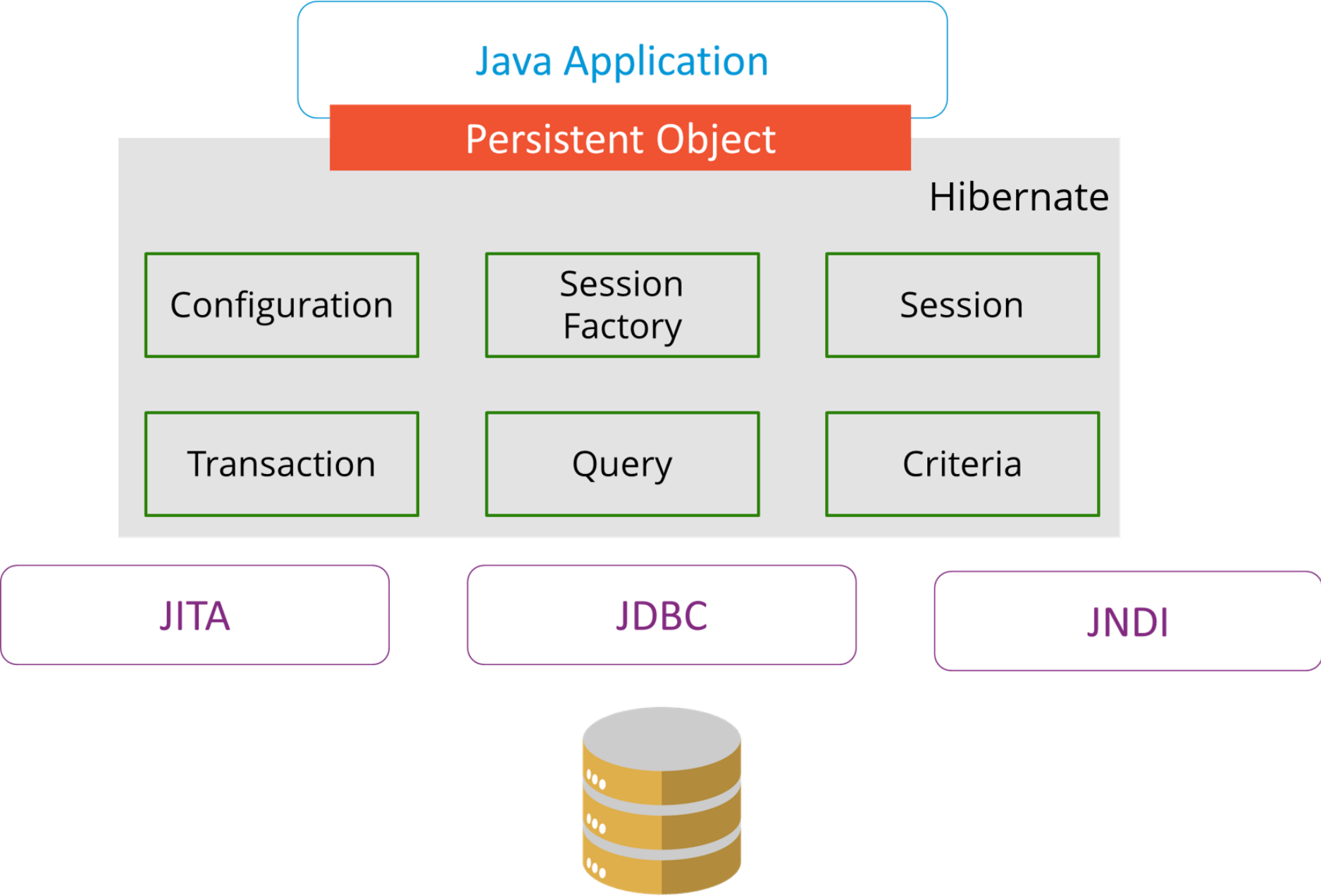


Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)



Q4 What are the differences between get and load methods?

The differences between get() and load() methods are given below.

No.	get()	load()
1)	Returns null if object is not found.	Throws ObjectNotFoundException if object is not found.
2)	get() method always hit the database.	load() method doesn't hit the database.
3)	It returns real object not proxy.	It returns proxy object.
4)	It should be used if you are not sure about the existence of instance.	It should be used if you are sure that instance exists.


Q5 What are the advantages of Hibernate over JDBC?

Some of the important advantages of Hibernate framework over JDBC are:

1. Hibernate removes a lot of boiler-plate code that comes with JDBC API, the code looks more cleaner and readable.
2. Hibernate supports inheritance, associations and collections. These features are not present with JDBC API.
3. Hibernate implicitly provides transaction management, in fact most of the queries can't be executed outside transaction. In JDBC API, we need to write code for transaction management using commit and rollback.
4. JDBC API throws SQLException that is a checked exception, so we need to write a lot of try-catch block code. Most of the times it's redundant in every JDBC call and used for transaction management. Hibernate wraps JDBC exceptions and throw JDBCException or HibernateException un-checked exception, so we don't need to write code to handle it. Hibernate built-in transaction management removes the usage of try-catch blocks.
5. Hibernate Query Language (HQL) is more object oriented and close to java programming language. For JDBC, we need to write native sql queries.
6. Hibernate supports caching that is better for performance, JDBC queries are not cached hence performance is low.

Subscribe to our Newsletter, and get personalized recommendations.

✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

Java Interview Questions: JSP

Q1. What are the life-cycle methods for a jsp?

Method	Description
public void jspInit()	It is invoked only once, same as init method of servlet.
public void _jspService(ServletRequest request,ServletResponse)throws ServletException,IOException	It is invoked at each request, same as service() method of servlet.
public void jspDestroy()	It is invoked only once, same as destroy() method of servlet.

Q2. What are the JSP implicit objects?

JSP provides 9 implicit objects by default. They are as follows:

Object	Type
1) out	JspWriter
2) request	HttpServletRequest
3) response	HttpServletResponse
4) config	ServletConfig
5) session	HttpSession
6) application	ServletContext
7) pageContext	PageContext
8) page	Object
9) exception	Throwable

Q3. What are the differences between include directive and include action?

include directive	include action
The include directive includes the content at page translation time.	The include action includes the content at request time.
The include directive includes the original content of the page so page size increases at runtime.	The include action doesn't include the original content rather invokes the include() method of Vendor provided class.
It's better for static pages.	It's better for dynamic pages.

Q4. How to disable caching on back button of the browser?


```
<%
response.setHeader("Cache-Control","no-store");
response.setHeader("Pragma","no-cache");
response.setHeader ("Expires", "0");           //prevents caching at the proxy server
%>
```


Q5. What are the different tags provided in JSTL?

There are 5 type of JSTL tags.

- 1. core tags
- 2. sql tags
- 3. xml tags
- 4. internationalization tags

Subscribe to our Newsletter, and get personalized recommendations. ✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

```

1  Cookie mycook = new Cookie("name1", "value1");
2
3  response.addCookie(mycook1);
4
5  Cookie killmycook = new Cookie("mycook1", "value1");
6
7  killmycook . set MaxAge ( 0 );
8
9  killmycook . set Path ("/");
10
11 killmycook . addCookie ( killmycook 1 );

```

Q8. Explain the jspDestroy() method.

jspDestry() method is invoked from javax.servlet.jsp.JspPage interface whenever a JSP page is about to be destroyed. Servlets destroy methods can be easily overridden to perform cleanup, like when closing a database connection.

Q9. How is JSP better than Servlet technology?

JSP is a technology on the server's side to make content generation simple. They are document centric, whereas servlets are programs. A Java server page can contain fragments of Java program, which execute and instantiate Java classes. However, they occur inside HTML template file. It provides the framework for development of a Web Application.

Q10. Why should we not configure JSP standard tags in web.xml?

We don't need to configure JSP standard tags in web.xml because when container loads the web application and find TLD files, it automatically configures them to be used directly in the application JSP pages. We just need to include it in the JSP page using taglib directive.

In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka!

Exception and Thread Java Interview Questions

Q1. What is difference between Error and Exception?

An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. These JVM errors you can not repair them at runtime. Though error can be caught in catch block but the execution of application will come to a halt and is not recoverable.

While exceptions are conditions that occur because of bad input or human error etc. e.g. FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference. In most of the cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.

Q2. How can you handle Java exceptions?

There are five keywords used to handle exceptions in java:

1. try
2. catch
3. finally
4. throw
5. throws

Q3. What are the differences between Checked Exception and Unchecked Exception?

Checked Exception

- The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions.
- Checked exceptions are checked at compile-time.

Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)

Final:

Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed. Let's take a look at the example below to understand it better.

```
1 | class FinalVarExample {
2 |     public static void main( String args[])
3 |     {
4 |         final int a=10;    // Final variable
5 |         a=50;              //Error as value can't be changed
6 |     }

```

Finally

Finally is used to place important code, it will be executed whether exception is handled or not. Let's take a look at the example below to understand it better.

```
1 | class FinallyExample {
2 |     public static void main(String args[]){
3 |         try {
4 |             int x=100;
5 |         }
6 |         catch(Exception e) {
7 |             System.out.println(e);
8 |         }
9 |         finally {
10 |             System.out.println("finally block is executing");}
11 |     }}
12 | }

```

Finalize

Finalize is used to perform clean up processing just before object is garbage collected. Let's take a look at the example below to understand it better.

```
1 | class FinalizeExample {
2 |     public void finalize() {
3 |         System.out.println("Finalize is called");
4 |     }
5 |     public static void main(String args[])
6 |     {
7 |         FinalizeExample f1=new FinalizeExample();
8 |         FinalizeExample f2=new FinalizeExample();
9 |         f1= NULL;
10 |        f2=NULL;
11 |        System.gc();
12 |    }
13 | }

```

Q5. What are the differences between throw and throws?

throw keyword	throws keyword
Throw is used to explicitly throw an exception.	Throws is used to declare an exception.
Checked exceptions can not be propagated with throw only.	Checked exception can be propagated with throws.
Throw is followed by an instance.	Throws is followed by class.
Throw is used within the method.	Throws is used with the method signature.
You cannot throw multiple exception	You can declare multiple exception e.g. public void method()throws IOException,SQLException.


Q6. What is exception hierarchy in java?

The hierarchy is as follows:

Throwable is a parent class of all Exception classes. There are two types of Exceptions: Checked exceptions and

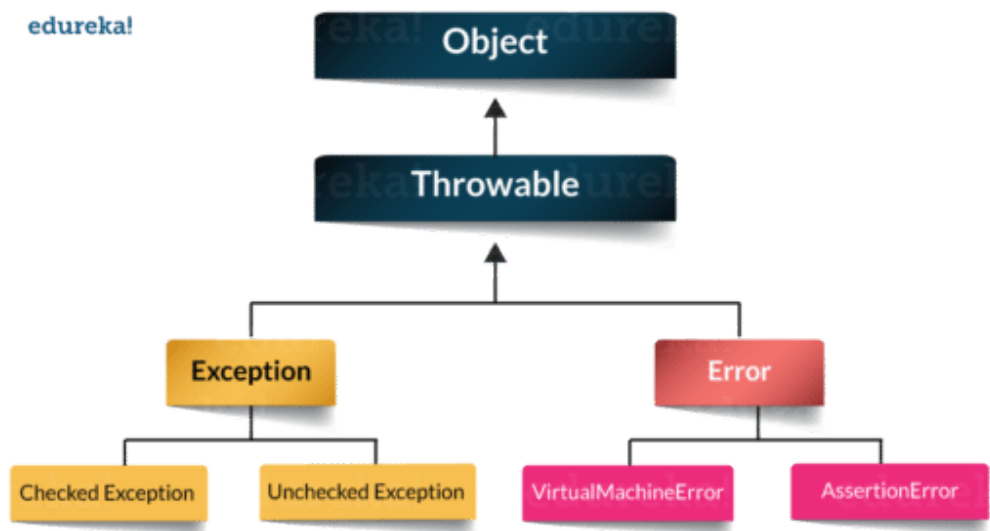
Subscribe to our Newsletter, and get personalized recommendations.

✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)



Java, J2EE & SOA Certification Training

[Weekday / Weekend Batches](#)

[See Batch Details](#)

Q7. How to create a custom Exception?

To create you own exception extend the Exception class or any of its subclasses.

- class New1Exception extends Exception { } // this will create Checked Exception
- class NewException extends IOException { } // this will create Checked exception
- class NewException extends NullPonterExcpetion { } // this will create UnChecked exception

Q8. What are the important methods of Java Exception Class?


Exception and all of it's subclasses doesn't provide any specific methods and all of the methods are defined in the base class Throwable.

1. **String getMessage()** – This method returns the message String of Throwable and the message can be provided while creating the exception through it's constructor.
2. **String getLocalizedMessage()** – This method is provided so that subclasses can override it to provide locale specific message to the calling program. Throwable class implementation of this method simply use getMessage() method to return the exception message.
3. **Synchronized Throwable getCause()** – This method returns the cause of the exception or null id the cause is unknown.
4. **String toString()** – This method returns the information about Throwable in String format, the returned String contains the name of Throwable class and localized message.
5. **void printStackTrace()** – This method prints the stack trace information to the standard error stream, this method is overloaded and we can pass PrintStream or PrintWriter as argument to write the stack trace information to the file or stream.

Q9. What are the differences between processes and threads?

	Process	Thread
Definition	An executing instance of a program is called a process.	A thread is a subset of the process.
Communication	Processes must use inter-process communication to communicate with sibling processes.	Threads can directly communicate with other threads of its process.
Control	Processes can only exercise control over child processes.	Threads can exercise considerable control over threads of the same process.
	Any change in the parent process	Any change in the main thread may affect the

Subscribe to our Newsletter, and get personalized recommendations. ✕

 Sign up with Google

 Signup with Facebook

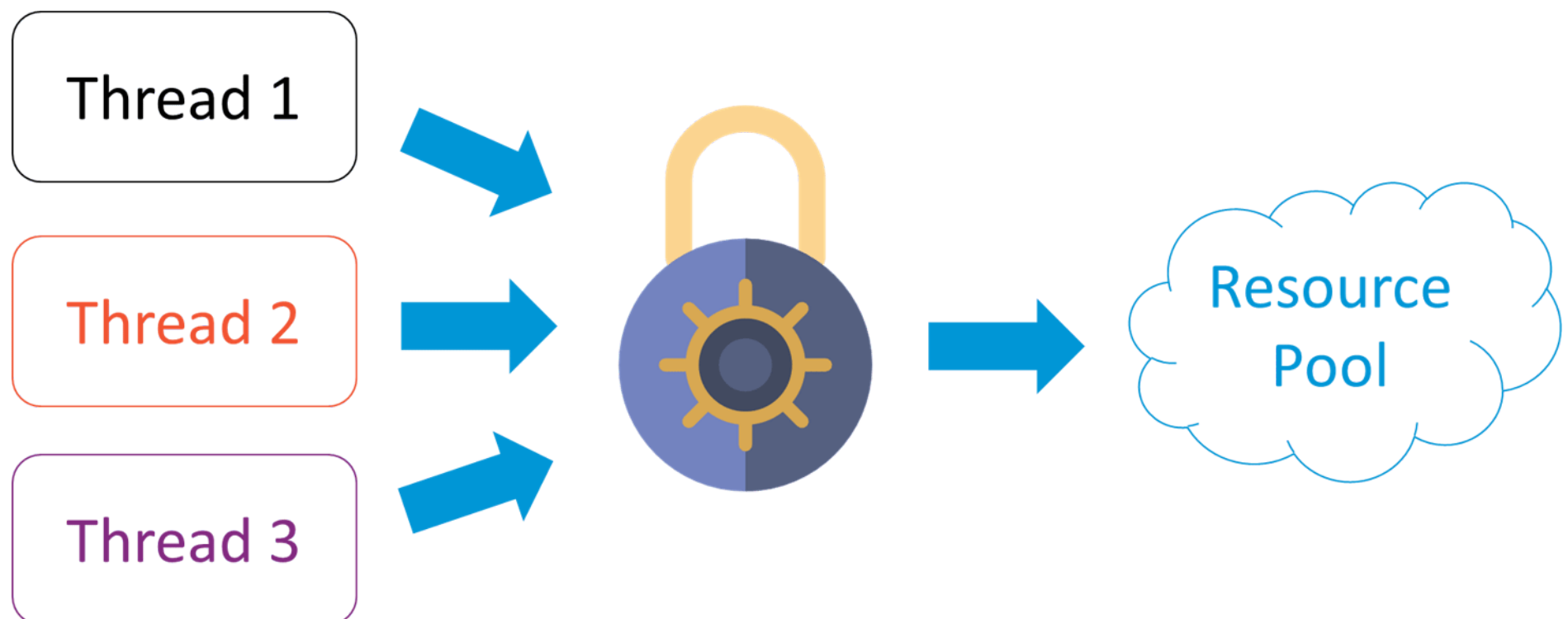
Already have an account? [Sign in.](#)

Finally block is a block which always executes a set of statements. It is always associated with a try block regardless of any exception that occurs or not.

Yes, finally will not be executed if the program exits either by calling `System.exit()` or by causing a fatal error that causes the process to abort.

Q11. What is synchronization?

Synchronization refers to multi-threading. A synchronized block of code can be executed by only one thread at a time. As Java supports execution of multiple threads, two or more threads may access the same fields or objects. Synchronization is a process which keeps all concurrent threads in execution to be in sync. Synchronization avoids memory consistency errors caused due to inconsistent view of shared memory. When a method is declared as synchronized the thread holds the monitor for that method's object. If another thread is executing the synchronized method the thread is blocked until that thread releases the monitor.



Q12. Can we write multiple catch blocks under single try block?

Yes we can have multiple catch blocks under single try block but the approach should be from specific to general. Let's understand this with a programmatic example.

```

1 public class Example {
2     public static void main(String args[]) {
3         try {
4             int a[] = new int[10];
5             a[10] = 10/0;
6         }
7         catch(ArithmeticException e)
8         {
9             System.out.println("Arithmetic exception in first catch block");
10        }
11        catch(ArrayIndexOutOfBoundsException e)
12        {
13            System.out.println("Array index out of bounds in second catch block");
14        }
15        catch(Exception e)
16        {
17            System.out.println("Any exception in third catch block");
18        }
19    }
  
```

Q13. What are the important methods of Java Exception Class?

Methods are defined in the base class `Throwable`. Some of the important methods of Java exception class are stated below.

1. **String getMessage()** – This method returns the message String about the exception. The message can be provided through its constructor.

Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)

In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

[Check Out Our Java Course](#)

So this brings us to the end of the Java interview questions blog. The topics that you learnt in this Java Interview Questions blog are the most sought-after skill sets that recruiters look for in a Java Professional. These set of Java Interview Questions will definitely help you ace your job interview. **Good luck with your interview!**

Check out the [Java training](#) by Edureka, a trusted online learning company with a network of more than 250,000 satisfied learners spread across the globe. We are here to help you with every step on your journey, for becoming a besides this java interview questions, we come up with a curriculum which is designed for students and professionals who want to be a Java Developer. The course is designed to give you a head start into Java programming and train you for both core and advanced Java concepts along with various Java frameworks like Hibernate & Spring.

Got a question for us? Please mention it in the comments section of this “Java Interview Questions” and we will get back to you as soon as possible.

Recommended articles for you



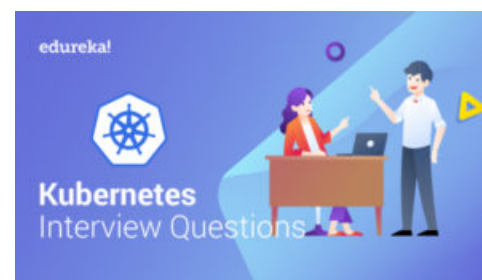
[Top 55 Blockchain Interview Questions You Must Prepare In 2019](#)

[Read Article](#)



[Top 50 PHP Interview Questions You Must Prepare in 2019](#)

[Read Article](#)



[Top 50 Kubernetes Interview Questions You Must Prepare In 2019](#)

[Read Article](#)



[Frequently Asked Questions Interview Questions](#)

[Read Article](#)

<>

Comments

37 Comments

Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)

40 Comments

https://www.edureka.co/blog/

1

Login

Recommend

11

Tweet

Share

Sort by Best

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

- Bhargav Patel

a year ago

Hi.
Section OOPS Java Interview Questions:
Q5 example is not valid in terms of syntax you missed curly braces.

14 ^ | v • Reply • Share >
- EdurekaSupport

Mod → Bhargav Patel • a year ago

Hey Bhargav, this was mostly probably a typo. We regret the mistake and thank you for pointing it out. Do browse through our other blogs and let us know how you liked it. Cheers :)

1 ^ | v • Reply • Share >
- Janine

9 months ago

Are you familiar with helpful books with small tasks which are used by recruiters during a programming interview?

2 ^ | v • Reply • Share >
- Chloe Jeffey

→ Janine • 9 months ago

"TOP 30 Java Interview Coding Tasks" by Matthew Urban
"Coding Interview Questions" by Narasimha Karumanchi
"Elements of Programming Interviews in Java" by Adnan Aziz, Tsung-Hsien Lee, Amit Prakash

^ | v • Reply • Share >
- Patrick Roth

→ Chloe Jeffey • 8 months ago

I have read all three books you recommend, in my opinion. The book "TOP 30 Java Interview Coding Tasks" covers only the most popular topics. "Coding Interview Questions" is too complicated for me. The volume of "Elements of Programming Interviews in Java" is too high...

^ | v • Reply • Share >
- Show more replies
- Swetha

5 months ago

Thank so much very useful

1 ^ | v • Reply • Share >
- thank u soo much !!!!!

8 months ago

It was really helpful ! may god bless the people involved in making this succesful !

1 ^ | v • Reply • Share >
- Jame Narongsak

8 months ago

Nice article. Thank for your dedicate to make all of this.

1 ^ | v • Reply • Share >
- Shimeles Tessema

9 months ago

In Question 10, describing the difference between heap and stack, the last comparison is not totally accurate. starting from Java 6 we have something called Escape Analysis made by the JVM. This is not only primitives and references are stored but also Objects that has no hope of being referenced outof scope are stored in the Stack Memory. This is efficient because the garbage collector shouldn't care about local variables that end their lifetime in side the same scope.

1 ^ | v • Reply • Share >

Subscribe to our Newsletter, and get personalized recommendations.

Sign up with Google

Signup with Facebook

Already have an account? [Sign in.](#)

**EdurekaSupport** • a year ago

Hey Neha, thak you for pointing this out. We will definitely look into this. Cheers :)

^ | v • Reply • Share ›

**Red Rajesh** • 2 years ago

explain very well excepted with image good

1 ^ | v • Reply • Share ›

**Will Smith** • 9 days agoI think that could be helpful <https://hubpages.com/techno...>

^ | v • Reply • Share ›

**Buddhi Prabhath** • 12 days ago

Q10. How to integrate Spring and Hibernate Frameworks?

JPA will be a good choice, since it uses Spring Data and Hibernate, implementing CRUD repository will make it easier to use basic CRUD methods like save, update, findall, delete

^ | v • Reply • Share ›

**Buddhi Prabhath** • 15 days ago

The OOP section question regarding the Abstract classes vs Interfaces

It's not true anymore with Java 1.8, that Interface also can have default methods same as Abstract classes

^ | v • Reply • Share ›

**Buddhi Prabhath** • 15 days ago

For the Question no. 7, basic questions, regarding Singleton class

1. Singleton class should have private constructor, this is given but also

2. It should have a "public" "static" method to get the Singleton object

^ | v • Reply • Share ›

**misha** • a month ago

Thank you ,

This all Questions are really heplful,keep sharing.

<https://www.exltech.in/java...>

^ | v • Reply • Share ›

**pagadala saidulu** • a month ago

Good Collection. Most Useful Java Interview Questions.

<https://www.winpathit.com/b...>

^ | v • Reply • Share ›

**Akshaya S** • 2 months agoThis site is awesome. <http: kosmiktechnologies.com="" java-training-in-hyderabad="" <="" java="">explanation is given is so neat, easy to understand. java concepts are explained nicely with easy of examples.

^ | v • Reply • Share ›

**bestinterviewquestion** • 3 months agoreally helpful post @[BestInterviewQuestion](#)

^ | v • Reply • Share ›

**Akash** • 3 months ago

In case of following question, the answer needs a little update.

Q5. Can you override a private or static method in Java?

Update. Private methods are actually supports concept of Method Hiding, but if you try overriding with Static method, you will get compilation error. As static method do not support method hiding concept, so beware of Static!

^ | v • Reply • Share ›

**Daniel** • 3 months ago

OOPS Java Interview Questions Q3: Since Java 8 you actually can have default code in interfaces.

^ | v • Reply • Share ›

Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google




Signup with Facebook

Already have an account? [Sign in.](#)


where i think that agency took most of their questions - <https://www.bitdegree.org/t...> . It's like 95% were taken from there maybe they just picked questions straight from tutorial websites ha ha :)

^ | v • Reply • Share ›

 **Swetha** • 5 months ago


Thank you keep writing

^ | v • Reply • Share ›

 **kamran** • 5 months ago

Missing static and non-static differences

^ | v • Reply • Share ›

 **V** • 5 months ago

The content has a lot of bugs.

This Code : (== vs .equals())

```
public class Equaltest {


    public static void main(String[] args) {
        String str1= new String("gfg");
        String str2= new String("gfg");
        if(str1 == str2){
            System.out.println("String 1 == String 2 is true");
        }
        else
            System.out.println("String 1 == String 2 is false");

        String str3 = str2;

        if( str2 == str3)
        {
```


see more

^ | v • Reply • Share ›

 **Greshma Liz George** • 6 months ago


In qn 18, isnt the parent method foo supposed to be hidden? so that all outputs belong to child class..

^ | v • Reply • Share ›

 **EdurekaSupport** Mod ➔ Greshma Liz George • 5 months ago


Hey Greshma, we would definitely like to help. May we know which question are you talking about? We can't find a qn18. Thanks!

^ | v • Reply • Share ›

 **prashant patil** • 7 months ago


This site is awesome. explanation is given is so neat, easy to understand. java concepts are explained nicely with easy of examples.

^ | v • Reply • Share ›

 **EdurekaSupport** Mod ➔ prashant patil • 7 months ago


Hey Prashant, thanks for the compliment. We are glad you loved the video. Do browse through our other blogs too. Cheers!

3 ^ | v • Reply • Share ›

 **K** • 10 months ago

Q3 1st and 3rd row are wrong. Interfaces have been able to provide code via default methods for years (Java 8, 2014).

^ | v • Reply • Share ›


 **Mirror, mirror, on the wall** ➔ K • 8 months ago

This isn't the the only one that's wrong. This article seems outdated and sometimes even misleading. For exmample, to create a custom Unchecked exception, extend RuntimeException (not NullPointerException as the author of this article suggests).

^ | v • Reply • Share ›


Subscribe to our Newsletter, and get personalized recommendations.

✕

 Sign up with Google

 Signup with Facebook

Already have an account? [Sign in.](#)

- 

what are the modifiers applicable in java main method


^

|

▼

• Reply

• Share

›
- 

Johny Herl

• a year ago

The Spring Framework is a lightweight framework for developing Java enterprise applications. It provides high performing, easily testable and reusable code. Spring handles the infrastructure as the underlying framework so that you can focus on your application.Spring is modular in design, handling and linking of individual components so much easier. Spring implements Model View Container(MVC) design pattern.

[spring mvc form example](#)


^

|

▼

• Reply

• Share

›
- 

Red Rajesh

• 2 years ago

thank you need sum advance concept

^

|

▼

• Reply


• Share

›

ALSO ON [HTTPS://WWW.EDUREKA.CO/BLOG/](https://www.edureka.co/blog/)


How to Become an Artificial Intelligence Engineer

1 comment • a month ago

 **Lela** — A getaway to the most exciting places in the entire world with all your family members and a lovely house you would like to purchase. Is your current ...


Ethereum Smart Contract – How to execute a Smart Contract?

1 comment • 2 months ago

 **ElvisPresley** — Really amazing article. We get brief explanation about how smart contract works. Surely I will share this article.If anyone having an idea to ...

Support Vector Machine In R | Using SVM To Predict Heart Diseases

1 comment • 3 months ago


 **Thiha kyaw** — How to preprocessing sale datasets for svm to predict customer ehavior?

#IndiaITRepublic – Top 10 Facts about Cognizant – India

1 comment • 3 months ago

 **Rajesh Kothuru** — Great👍

 [Subscribe](#)

 [Add Disqus to your site](#)[Add Disqus](#)[Add](#)

 [Disqus' Privacy Policy](#)[Privacy Policy](#)[Privacy Policy](#)[Privacy](#)

Trending Courses in Programming & Frameworks



[Java, J2EE & SOA Certification Training](#)

 [29k Enrolled Learners](#)

 [Weekend](#)

 [Live Class](#)

[Reviews](#)

★★★★★

4 (11250)



[Python Scripting Certification Training](#)

 [5k Enrolled Learners](#)

 [Weekend](#)

 [Self Paced](#)

[Reviews](#)

★★★★★

5 (1850)



[Spring Framework Certification Training](#)

 [7k Enrolled Learners](#)

 [Weekend](#)

 [Live Class](#)

[Reviews](#)

★★★★★

5 (2500)



[Comprehensive Java Certification Training](#)

 [15k Enrolled Learners](#)

 [Weekend/Weekday](#)

 [Self Paced](#)

[Reviews](#)

★★★★★


5 (5650)

⏪


Browse Categories

- Artificial Intelligence
- BI and Visualization
- Big Data
- Blockchain
- Cloud Computing
- Cyber Security
- Data Science

Subscribe to our Newsletter, and get personalized recommendations.




Sign up with Google




Signup with Facebook

Already have an account? [Sign in.](#)

Subscribe to our Newsletter, and get personalized recommendations. ✕



Sign up with Google



Signup with Facebook

Already have an account? [Sign in.](#)