1. What is the right data type to represent a price in Java?

Answer: BigDecimal, if memory is not a concern and Performance, is not critical, otherwise double with predefined precision.

1. What is a class in Java?

Answer: Java encapsulates the codes in various classes which define new data types. These new data types are used to create objects.

3. What is a JVM?

Answer: JVM is Java Virtual Machine which is a run time environment for the compiled java class files.

4. Explain method overloading?

Answer: When a Java program contains more than one methods with the same name but different properties, then it is called method overloading.

5. Does Java support multiple inheritances?

Answer: Java doesn’t support multiple inheritances.

6. What restrictions are placed on the location of a package statement within a source code file?

Answer: A package statement must appear as the first line in a source code file (eliminating blank lines and comments).

Read Out: Common Interview Mistakes

7. What is a transient variable?

Answer: A transient variable is a variable that may not be serialized.

8. Is null a keyword?

Answer:

No, the null is not a keyword.

9. What’s new with the stop(), suspend() and resume() methods in JDK 1.2 ?

Answer:

These methods have been deprecated in JDK 1.2.

10. What method is used to specify a container’s layout?

Answer: The setLayout() method is used to specify a container’s layout.

11. What is the immediate superclass of the Applet class?

Answer: The Panel class is the immediate superclass of the Applet class.

12. Can we rethrow the same exception from catch handler?

Answer: Yes, we can rethrow the same exception from our catch handler. If we want to rethrow checked exception

From a catch block we need to declare that exception.

13. what value is a variable of the String type automatically initialized?

Answer: The default value of a String type is null.

14. When a thread blocks on I/O, what state does it enter?

Answer: When it blocks on I/O, A thread enters the waiting state.

15. Which containers use a Flow Layout as their default layout?

Answer: The Panel and Applet classes use the Flow Layout as their default layout.

16. What modifiers may be used with an inner class that is a member of an outer class?

Answer: A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

17. Which java. Util classes and interfaces support event handling?

Answer: The Event Object class and the Event Listener interface support event processing. ( oracle apex training online )

18. What is the Vector class?

Answer: The term Vector class provides the ability to implement a growable array of objects.

19. What is the difference between the >> and >>> operators?

Answer: The >> operator carries the sign bit when shifting right while the >>> zero-fills bits that have been shifted out.

20. What is a native method?

Answer: A native method is a method that is applied in a language other than Java.

21. What value does read Line() return when it has reached the end of a file?

Answer:

The readLine() method returns null when it has reached the end of a file.

22. What is clipping?

Answer: Clipping is the process of confining paint operations to a limited area or shape.

23. Can a for statement loop indefinitely?

Answer: Yes, a for statement can loop indefinitely. For example, consider the following: for(;;)

24. Explain Java Coding standards for Methods?

Answer:

1) Method names should start with small letters.

2) Method names are usually verbs

3) If a method contains multiple words, every inner word should start with an uppercase letter.

Ex : toString()

4) Method name must be combination of verb and noun

Ex : getCarName(),getCarNumber()

25. Explain Java Coding Standards for Constants ?

Answer:

Constants in java are created using static and final keywords.

1) Constants contain only uppercase letters.

2) If the constant name is a combination of two words it should be separated by an underscore.

3) Constant names are usually nouns.

Ex:MAX\_VALUE, MIN\_VALUE, MAX\_PRIORITY, MIN\_PRIORITY

26. What is synchronization and why is it important?

Answer: The term synchronization is the ability to control the access of multiple threads to shared resources. And it is important because, without it, it is not possible for one thread to modify a shared object while another thread is in the process of using or updating that object’s value. This often leads to major errors.

27. Explain Java Coding Standards for variables?

Answer:

1) Variable names should start with small letters.

2) Variable names should be nouns

3) Short meaningful names are recommended.

4) If there are multiple words every inner world should start with Uppercase character.

Ex : string,value,empName,MEP salary

28. Name three Component subclasses that support painting?

Answer:

The Canvas, Frame, Panel, and Applet classes support painting.

29. What is the difference between JDK and JVM?

Answer: we’re presenting the difference between JDK and JVM in tabular format, take a look…

JDK JVM

Full-Form: Java Development Kit Full Form: Java Virtual Machine

For Development Purpose

To execute the java programs

It provides all the tools, executables and binaries required to compile, debug and execute a Java Program The execution part is handled by JVM to provide machine independence.

30. Why Java doesn’t support multiple inheritances?

Answer: Because of “Diamond Problem”, Java doesn’t support multiple inheritances in classes.

31. Why Java is not a pure Object Oriented language?

Answer: Java supports primitive types such as int, byte, short, long, etc that why it is not said to be a pure object-oriented language.

32. What are the access modifiers?

Answer: Java provides three access controls such as public, private and protected access modifier. When none of these are used, it’s called default access modifier.

33. Can we overload the main method?

Answer:

Yes, we can overload the main method with syntax as public static void main(String args[]).

34. What is the method in java?

Answer: It contains the executable body that can be applied to the specific object of the class.

The method includes method name, parameters or arguments and return type and a body of executable code.

Syntax : type methodName(Argument List){

Ex: public float add(int a, int b, int c)

Methods can have multiple arguments. Separate with commas when we have multiple arguments.

Thrown in the method are instances of their subclass.

35. Can we use catch statement for checked exceptions?

Answer: If there is no chance of raising an exception in our code then we can’t declare catch block for handling

Checked exceptions. This raises a compile-time error if we try to handle checked exceptions when there is

No possibility of causing an exception.

36. Explain a situation where finally block will not be executed?

Answer: Finally, the block will not be executed whenever JVM shutdowns. If we use system.exit(0) in try statement

Finally block if present will not be executed.

37. Explain about the main() method in java?

Answer: The main () method is the starting point of execution for all java applications.

Public static void main(String[] args) {}

String args[] are an array of string objects we need to pass from command line arguments.

Every Java application must have at least one main method. Company

38. What is constructor in java?

Answer: A constructor is a special method used to initialize objects in the java.

We use constructors to initialize all variables in the class when an object is created. As and when an object

Is created it is initialized automatically with the help of constructor in java.

We have two types of constructors

Default Constructor

Parameterized Constructor

39. How can we find the actual size of an object on the heap?

Answer: In Java, there is no way to find out the actual size of an object on the heap.

40. In how many ways we can do synchronization in java?

Answer:

There are two ways to do synchronization in java:

1) Synchronized methods

2) Synchronized blocks

To do synchronization we use the synchronized keyword.

41. Explain about Automatic type conversion in java?

Answer:

Java automatic type conversion is done if the following conditions are met:

1. When two types are compatible

Ex: int, float

Int can be assigned directly to float variable.

1. Destination type is larger than source type.

Ex: int, long.

Int can be assigned directly to long .Automatic type conversion takes place if int is assigned to long

Because long is larger datatype than int.

Widening Conversion comes under Automatic type conversion.

42. In how many ways we can do exception handling in java?

Answer:

We can handle exceptions in either of the two ways :

1) By specifying a try-catch block where we can catch the exception.

2) Declaring a method with throws clause.

43. What does null mean in java?

Answer:

When a reference variable doesn’t point to any value it is assigned null.

Example: Employee employee;

In the above example employee object is not instantiate so it is pointed nowhere.

44. Can we define a package statement after the import statement in java?

Answer: We can’t define a package statement after the import statement in java. A package statement must be the first statement in the source file. We can have commented before the package statement.

45. Explain where variables are created in memory?

Answer: When we declare variables are created in the stack. So when the variable is out of scope those variables get garbage collected.

46. When do we use synchronized blocks and advantages of using synchronized blocks?

Answer: If very few lines of code require synchronization then it is recommended to use synchronized blocks. The main advantage of synchronized blocks over synchronized methods is it reduces the waiting time of threads and improves performance of the system.

47. What is the difference between access specifiers and access modifiers in java?

Answer: In C++ we have access specifiers as public, private, protected and default and access modifiers as static, final. But there is no such division of access specifiers and access modifiers in java. In Java, we have access to modifiers and nonaccess modifiers.

Access Modifiers: public, private, protected, default

Non Access Modifiers: abstract, final, strip.

48. What access modifiers can be used for class?

Answer: We can use only two access modifiers for class public and default.

Public: A class with a public modifier can be visible

1) In the same class

2) In the same package subclass

3) In the same package nonsubclass

4) In the different package subclass

5) In the different package nonsubclass.

Default: A class with default modifier can be accessed

1) In the same class

2) In the same package subclass

3) In the same package nonsubclass

4) In the different package subclass

5) In the different package nonsubclass. ( )

49. Explain about abstract classes in java?

Answer: Sometimes we may come across a situation where we cannot provide implementation to all the methods in a class. We want to leave the implementation to a class that extends it. In such a case, we declare a class as abstract. To make a class abstract we use keyword abstract. Any class that contains one or more abstract methods is declared as abstract. If we don’t declare a class as abstract which contains abstract

Methods we get a compile-time error. We get the following error. “The type must be an abstract class to define abstract methods.” Signature; abstract class.

For example, if we take a vehicle class we cannot provide implementation to it because there may be two-wheelers, four-wheelers, etc. At that moment we make vehicle class abstract. All the common features of vehicles are declared as abstract methods in vehicle class. Any class which extends the vehicle will provide its method implementation. It’s the responsibility of subclass to provide the implementation.

The important features of abstract classes are:

1) Abstract classes cannot be instantiated.

2) An abstract class contains abstract methods, concrete methods or both.

3) Any class which extends abstract class must override all methods of an abstract class.

4) An abstract class can contain either 0 or more abstract methods.

Though we cannot instantiate abstract classes we can create object references. Through superclass

References, we can point to subclass.

50. Can we create a constructor in abstract class?

Answer: We can create a constructor in the abstract class, it doesn’t give any compilation error. But when we cannot

Instantiate class there is no use in creating a constructor for abstract class.

51. What are abstract methods in java?

Answer: An abstract method is a method which doesn’t have anybody. An abstract method is declared with

Keyword abstract and semicolon in place of the method body.

Signature : public abstract void ();

Ex : public abstract void get details();

It is the responsibility of subclass to provide implementation to an abstract method defined in the abstract class.

52. State some situations where exceptions may arise in java?

Answer:

1) Accessing an element that does not exist in the array.

2) Invalid conversion of number to string and string to a number.

(NumberFormatException)

1. The invalid casting of class

(Class cast Exception)

1. Trying to create an object for interface or abstract class

(Instantiation Exception)

53. What is an exception in java?

Answer:

In java, an exception is an object. Exceptions are created when abnormal situations arise in our program. Exceptions can be created by JVM or by our application code. All Exception classes are defined in java.lang. In other words, we can say Exception as a run time error.

54. What is an error in Java?

Answer: Error is the subclass of Throwable class in java. When errors are caused by our program we call that as Exception, but some times exceptions are caused due to some environmental issues such as running out of memory. In such cases, we can’t handle the exceptions. Exceptions which cannot be recovered are called as errors in java.

Ex: Out of memory issues.

55. What are the advantages of Exception handling in java?

Answer:

1) Separating normal code from exception handling code to avoid abnormal termination of the program.

2) Categorizing into different types of Exceptions so that rather than handling all exceptions with

Exception root class we can handle with specific exceptions. It is recommended to handle exceptions with

Specific Exception instead of handling with Exception root class.

3) Call stack mechanism: If a method throws an exception and it is not handled immediately, then that

Exception is propagated or thrown to the caller of that method. This propagation continues till it finds an

Appropriate exception handler, if it finds handler it would be handled otherwise program terminates

Abruptly.

56. In how many ways we can create threads in java?

Answer:

We can create threads in java by any of the two ways :

1) By extending Thread class

2) By implementing the Runnable interface.

57. Explain creating threads by implementing Runnable class?

Answer: This is the first and foremost way to create threads. By implementing the runnable interface and implementing

The run() method we can create a new thread.

Method signature : public void run()

Run is the starting point for execution for another thread within our program.

Example :

Public class MyClass implements Runnable {

@Override

Public void run()

58. When do we use synchronized methods in java?

Answer: If multiple threads try to access a method where the method can manipulate the state of the object, in such a scenario we can declare a method as synchronized.

59. Explain the importance of finally block in java?

Answer: Finally block is used for cleaning up of resources such as closing connections, sockets, etc. if try block executes with no exceptions then finally is called after try block without executing catch block. If there is an exception thrown in try block finally block executes immediately after the catch block. If an exception is thrown, finally block will be executed even if the no catch block handles the exception.

60. Can we catch more than one exception in a single catch block?

Answer: From Java 7, we can catch more than one exception with a single catch block. This type of handling reduces

The code duplication.

Note: When we catch more than one exception in a single catch block, catch parameter is implicitly final.

We cannot assign any value to catch parameter.

Ex : catch(ArrayIndexOutOfBoundsException || ArithmeticException e)

JSP

12.1 What is a JSP Page ?

A Java Server Page (JSP) is a text document that contains two types of text: static data and JSP elements. Static data can be

Expressed in any text-based format, such as HTML or XML. JSP is a technology that mixes static content with dynamically-

Generated content. See JSP example here.

12.2 How are the JSP requests handled ?

On the arrival of a JSP request, the browser first requests a page with a .jsp extension. Then, the Web server reads the request and

Using the JSP compiler, the Web server converts the JSP page into a servlet class. Notice that the JSP file is compiled only on

The first request of the page, or if the JSP file has changed.The generated servlet class is invoked, in order to handle the browser’s

Request. Once the execution of the request is over, the servlet sends a response back to the client. See how to get Request

Parameters in a JSP.

12.3 What are the advantages of JSP ?

The advantages of using the JSP technology are shown below:

• JSP pages are dynamically compiled into servlets and thus, the developers can easily make updates to presentation code.

• JSP pages can be pre-compiled.

• JSP pages can be easily combined to static templates, including HTML or XML fragments, with code that generates dynamic

Content.

• Developers can offer customized JSP tag libraries that page authors access using an XML-like syntax.

• Developers can make logic changes at the component level, without editing the individual pages that use the application’s logic.

12.4 What are Directives ? What are the different types of Directives available in

JSP ?

Directives are instructions that are processed by the JSP engine, when the page is compiled to a servlet. Directives are used to

Set page-level instructions, insert data from external files, and specify custom tag libraries. Directives are defined between < %@

And % >. The different types of directives are shown below:

• Include directive: it is used to include a file and merges the content of the file with the current page.

• Page directive: it is used to define specific attributes in the JSP page, like error page and buffer.

• Taglib: it is used to declare a custom tag library which is used in the page.

12.5 What are JSP actions ?

JSP actions use constructs in XML syntax to control the behavior of the servlet engine. JSP actions are executed when a JSP

Page is requested. They can be dynamically inserted into a file, re-use JavaBeans components, forward the user to another page,

Or generate HTML for the Java plugin.Some of the available actions are listed below:

• jsp:include – includes a file, when the JSP page is requested.

• jsp:useBean – finds or instantiates a JavaBean.

• jsp:setProperty – sets the property of a JavaBean.

• jsp:getProperty – gets the property of a JavaBean.

• jsp:forward – forwards the requester to a new page.

• jsp:plugin – generates browser-specific code.

12.6 What are Scriptlets ?

In Java Server Pages (JSP) technology, a scriptlet is a piece of Java-code embedded in a JSP page. The scriptlet is everything

Inside the tags. Between these tags, a user can add any valid scriplet.

12.7 What are Decalarations ?

Declarations are similar to variable declarations in Java. Declarations are used to declare variables for subsequent use in expres-

Sions or scriptlets. To add a declaration, you must use the sequences to enclose your declarations.

12.8 What are Expressions ?

A JSP expression is used to insert the value of a scripting language expression, converted into a string, into the data stream

Returned to the client, by the web server. Expressions are defined between <% =and %> tags.

12.9 What is meant by implicit objects and what are they ?

JSP implicit objects are those Java objects that the JSP Container makes available to developers in each page. A developer can

Call them directly, without being explicitly declared. JSP Implicit Objects are also called pre-defined variables.The following

Objects are considered implicit in a JSP page:

• application

• page

• request

• response

• session • exception • out • config • pageContex

Servlets

11.1 What is a Servlet ?

The servlet is a Java programming language class used to process client requests and generate dynamic web content. Servlets are

Mostly used to process or store data submitted by an HTML form, provide dynamic content and manage state information that

Does not exist in the stateless HTTP protocol.

11.2 Explain the architechure of a Servlet.

The core abstraction that must be implemented by all servlets is the javax.servlet.Servlet interface. Each servlet must implement

It either directly or indirectly, either by extending javax.servlet.GenericServlet or javax.servlet.http.HTTPServlet. Finally, each

Servlet is able to serve multiple requests in parallel using multithreading.

11.3 What is the difference between an Applet and a Servlet ?

An Applet is a client side java program that runs within a Web browser on the client machine. On the other hand, a servlet is a

Server side component that runs on the web server.An applet can use the user interface classes, while a servlet does not have a

User interface. Instead, a servlet waits for client’s HTTP requests and generates a response in every request.

11.4 What is the difference between GenericServlet and HttpServlet ?

GenericServlet is a generalized and protocol-independent servlet that implements the Servlet and ServletConfig interfaces. Those

Servlets extending the GenericServlet class shall override the service method. Finally, in order to develop an HTTP servlet for

Use on the Web that serves requests using the HTTP protocol, your servlet must extend the HttpServlet instead. Check Servlet

Examples here.

11.5 Explain the life cycle of a Servlet.

On every client’s request, the Servlet Engine loads the servlets and invokes its init methods, in order for the servlet to be

Initialized. Then, the Servlet object handles all subsequent requests coming from that client, by invoking the service method for

Each request separately. Finally, the servlet is removed by calling the server’s destroy method.

11.6 What is the difference between doGet() and doPost() ?

doGET: The GET method appends the name-value pairs on the request’s URL. Thus, there is a limit on the number of characters

and subsequently on the number of values that can be used in a client’s request. Furthermore, the values of the request are made

visible and thus, sensitive information must not be passed in that way.

doPOST: The POST method overcomes the limit imposed by the GET request, by sending the values of the request inside its

body. Also, there is no limitations on the number of values to be sent across. Finally, the sensitive information passed through a

POST request is not visible to an external client.

11.7 What is meant by a Web Application ?

A Web application is a dynamic extension of a Web or application server. There are two types of web applications: presentation-

Oriented and service-oriented. A presentation-oriented Web application generates interactive web pages, which contain various

Types of markup language and dynamic content in response to requests. On the other hand, a service-oriented web application

Implements the endpoint of a web service. In general, a Web application can be seen as a collection of servlets installed under a

Specific subset of the server’s URL namespace.

11.8 What is a Server Side Include (SSI) ?

Server Side Includes (SSI) is a simple interpreted server-side scripting language, used almost exclusively for the Web, and is

Embedded with a servlet tag. The most frequent use of SSI is to include the contents of one or more files into a Web page on a

Web server. When a Web page is accessed by a browser, the Web server replaces the servlet tag in that Web page with the hyper

Text generated by the corresponding servlet.

11.9 What is Servlet Chaining ?

Servlet Chaining is the method where the output of one servlet is sent to a second servlet. The output of the second servlet can

Be sent to a third servlet, and so on. The last servlet in the chain is responsible for sending the response to the client.

11.10 How do you find out what client machine is making a request to your servlet

?

The ServletRequest class has functions for finding out the IP address or host name of the client machine. getRemoteAddr() gets

The IP address of the client machine and getRemoteHost() gets the host name of the client machine. See example here.

11.11 What is the structure of the HTTP response ?

The HTTP response consists of three parts:

• Status Code: describes the status of the response. It can be used to check if the request has been successfully completed. In

Case the request failed, the status code can be used to find out the reason behind the failure. If your servlet does not return a

Status code, the success status code, HttpServletResponse.SC\_OK, is returned by default.

• HTTP Headers: they contain more information about the response. For example, the headers may specify the date/time after

Which the response is considered stale, or the form of encoding used to safely transfer the entity to the user. See how to retrieve

Headers in Servlet here.

• Body: it contains the content of the response. The body may contain HTML code, an image, etc. The body consists of the data

Bytes transmitted in an HTTP transaction message immediately following the headers.

11.12 What is a cookie ? What is the difference between session and cookie ?

A cookie is a bit of information that the Web server sends to the browser. The browser stores the cookies for each Web server

In a local file. In a future request, the browser, along with the request, sends all stored cookies for that specific Web server.The

Differences between session and a cookie are the following:

• The session should work, regardless of the settings on the client browser. The client may have chosen to disable cookies.

However, the sessions still work, as the client has no ability to disable them in the server side.

• The session and cookies also differ in the amount of information the can store. The HTTP session is capable of storing any

Java object, while a cookie can only store String objects.

11.13 Which protocol will be used by browser and servlet to communicate ?

The browser communicates with a servlet by using the HTTP protocol.

11.14 What is HTTP Tunneling ?

HTTP Tunneling is a technique by which, communications performed using various network protocols are encapsulated using

The HTTP or HTTPS protocols. The HTTP protocol therefore acts as a wrapper for a channel that the network protocol being

Tunneled uses to communicate. The masking of other protocol requests as HTTP requests is HTTP Tunneling.

11.15 What’s the difference between sendRedirect and forward methods ?

The sendRedirect method creates a new request, while the forward method just forwards a request to a new target. The previous

Request scope objects are not available after a redirect, because it results in a new request. On the other hand, the previous request

Scope objects are available after forwarding. Finally, in general, the sendRedirect method is considered to be slower compare to

The forward method.

11.16 What is URL Encoding and URL Decoding ?

The URL encoding procedure is responsible for replacing all the spaces and every other extra special character of a URL, into

Their corresponding Hex representation. In correspondence, URL decoding is the exact opposite procedure.

JDBC

9.1 What is JDBC ?

JDBC is an abstraction layer that allows users to choose between databases. JDBC enables developers to write database applica-

Tions in Java, without having to concern themselves with the underlying details of a particular database.

9.2 Explain the role of Driver in JDBC.

The JDBC Driver provides vendor-specific implementations of the abstract classes provided by the JDBC API. Each driver

Must provide implementations for the following classes of the java.sql package:Connection, Statement, PreparedStatement,

CallableStatement, ResultSet and Driver.

9.3 What is the purpose Class.forName method ?

This method is used to method is used to load the driver that will establish a connection to the database.

9.4 What is the advantage of PreparedStatement over Statement ?

PreparedStatements are precompiled and thus, their performance is much better. Also, PreparedStatement objects can be reused

With different input values to their queries.

9.5 What is the use of CallableStatement ? Name the method, which is used to

Prepare a CallableStatement.

A CallableStatement is used to execute stored procedures. Stored procedures are stored and offered by a database. Stored

Procedures may take input values from the user and may return a result. The usage of stored procedures is highly encouraged,

Because it offers security and modularity.The method that prepares a CallableStatement is the following: CallableStament.

Exception Handling

6.1 What are the two types of Exceptions in Java ? Which are the differences

Between them ?

Java has two types of exceptions: checked exceptions and unchecked exceptions. Unchecked exceptions do not need to be

Declared in a method or a constructor’s throws clause, if they can be thrown by the execution of the method or the constructor,

And propagate outside the method or constructor boundary. On the other hand, checked exceptions must be declared in a method

Or a constructor’s throws clause. See here for tips on Java exception handling.

6.2 What is the difference between Exception and Error in java ?

Exception and Error classes are both subclasses of the Throwable class. The Exception class is used for exceptional conditions

That a user’s program should catch. The Error class defines exceptions that are not excepted to be caught by the user program.

6.3 What is the difference between throw and throws ?

The throw keyword is used to explicitly raise a exception within the program. On the contrary, the throws clause is used to

Indicate those exceptions that are not handled by a method. Each method must explicitly specify which exceptions does not

Handle, so the callers of that method can guard against possible exceptions. Finally, multiple exceptions are separated by a

Comma.

6.4 What is the importance of finally block in exception handling ?

A finally block will always be executed, whether or not an exception is actually thrown. Even in the case where the catch

Statement is missing and an exception is thrown, the finally block will still be executed. Last thing to mention is that the finally

Block is used to release resources like I/O buffers, database connections, etc.

6.5 What will happen to the Exception object after exception handling ?

The Exception object will be garbage collected in the next garbage collection.

6.6 How does finally block differ from finalize() method ?

A finally block will be executed whether or not an exception is thrown and is used to release those resources held by the

Application. Finalize is a protected method of the Object class, which is called by the Java Virtual Machine (JVM) just before an

Object is garbage collected.

Garbage Collectors

5.1 What is the purpose of garbage collection in Java, and when is it used ?

The purpose of garbage collection is to identify and discard those objects that are no longer needed by the application, in order

For the resources to be reclaimed and reused.

5.2 What does System.gc() and Runtime.gc() methods do ?

These methods can be used as a hint to the JVM, in order to start a garbage collection. However, this it is up to the Java Virtual

Machine (JVM) to start the garbage collection immediately or later in time.

5.3 When is the finalize() called ? What is the purpose of finalization ?

The finalize method is called by the garbage collector, just before releasing the object’s memory. It is normally advised to release

Resources held by the object inside the finalize method.

5.4 If an object reference is set to null, will the Garbage Collector immediately free

The memory held by that object ?

No, the object will be available for garbage collection in the next cycle of the garbage collector.

5.5 What is structure of Java Heap ? What is Perm Gen space in Heap ?

The JVM has a heap that is the runtime data area from which memory for all class instances and arrays is allocated. It is created

At the JVM start-up. Heap memory for objects is reclaimed by an automatic memory management system which is known as a

Garbage collector. Heap memory consists of live and dead objects. Live objects are accessible by the application and will not

Be a subject of garbage collection. Dead objects are those which will never be accessible by the application, but have not been

Collected by the garbage collector yet. Such objects occupy the heap memory space until they are eventually collected by the gc

5.6 What is the difference between Serial and Throughput Garbage collector ?

The throughput garbage collector uses a parallel version of the young generation collector and is meant to be used with applica-

Tions that have medium to large data sets. On the other hand, the serial collector is usually adequate for most small applications

(those requiring heaps of up to approximately 100MB on modern processors).

5.7 When does an Object becomes eligible for Garbage collection in Java ?

A Java object is subject to garbage collection when it becomes unreachable to the program in which it is currently used.

5.8 Does Garbage collection occur in permanent generation space in JVM ?

Garbage Collection does occur in PermGen space and if PermGen space is full or cross a threshold, it can trigger a full garbage

Collection. If you look carefully at the output of the garbage collector, you will find that PermGen space is also garbage collected.

What is Hibernate?

We can categorize Hibernate under the Object Relational Mapping (ORM), which features the mapping of Java classes to data tables and mapping from Java data types to SQL data types.

Hibernate is based on Java Virtual Machine and written in Java.

Hibernate provides data query and retrieval facilities.

Hibernate recommended in enterprise applications for database operations.

It is one of the highly used ORM tools for Java applications.

What is Query level cache in Hibernate?

We can implement a cached query in Hibernate that results in sets and integrates tightly with the second-level cache. But it is optional, and it requires two cache regions that can hold the cached query results and also the timestamps whenever a table is updated. We can use it only for the queries that run frequently containing the same parameters.

Example

<property name=”hibernate.cache.use\_query\_cache”>True</property>

Query query = session.createQuery(“from Employee”);

Query.setCacheable(true);

Query.setCacheRegion(“ALL\_EMP”);

Which annotation should be used to declare a class as a Hibernate bean?

We can use @Entity annotation to declare a class as an entity.

Example

@Entity

@Table(name=”post”)

Public class Post{

String title;

String description;

}

What is Lazy loading in Hibernate?

In lazy loading, we can load objects on a requirement basis. Lazy loading is by default enabled from Hibernate 3.0 so that the child objects not charged while the parent is loaded.

What is the first level cache in Hibernate?

It is a session cache and mandatory cache. It is from first level cache through which all the requests must pass. The session object stores an object under its control before committing it to the database.

What are Inheritance Mapping Strategies available in Hibernate?

Hibernate has three ways of inheritance mapping, as listed below.

Table per concrete class

Table per hierarchy

Table per subclass

Describe the Criteria object in Hibernate?

We can use criteria objects to create and execute object-oriented Queries to retrieve the objects.

Explain the Session object in Hibernate?

It is used to get a connection with a database. A session object is created to instantiate an interaction with the database, whereas the persistent objects are retrieved using a session object. The session objects are not thread-safe so that they be created and destroyed as per the requirement.

What is a One-to-One association in Hibernate?

A one to one association is similar to many to one association, and only one difference is that it must be set as a unique one. We can use many to one element to define one to one association.

A name attribute must set in the parent class, and the column attribute is used to set a column name in the parent table for the defined variable. It is unique so that only one object gets associated with another.

What is the use of the Hibernate Session merge() call?

We can use the Hibernate Session merge() to update existing values; however, this method creates a copy from the passed entity object and returns it. The returned object is part of the persistent context and tracked for any changes; the given object is not tracked. For example, a program, read Hibernate merge.

What will happen if a user doesn’t have no-args constructor in Entity bean?

Hibernate uses Reflection API to create an instance of Entity beans while calling get() or load() method. The method Class.newInstance() is used for this, and it requires no-args constructor. So if you won’t have no-args constructor in entity beans, hibernate will fail to instantiate it, and you will get HibernateException.

What are the collection types in Hibernate?

The Five collection types in hibernate used for one-to-many relationship mappings are as follows.

Bag

Set

List

Array

Map

How to implement relationships in hibernate?

We can quickly implement relationships like One to One Mapping, One to Many Mapping, and Many to Many Mappingusing JPA annotations as well as XML based configurations.

How to use JNDI DataSource with the Hibernate framework?

It’s always useful to allow a servlet container to manage the connection pool as the user so that we define JNDI resource for DataSource, and we can use it in the web application. It’s straightforward to use in Hibernate, all we need is to remove all the database-specific properties and use the below feature to provide the JNDI DataSource name.

Example

**What is a One-to-One association in Hibernate?**

* A one to one association is similar to many to one association, and only one difference is that it must be set as a unique one. We can use many to one element to define one to one association.
* A name attribute must set in the parent class, and the column attribute is used to set a column name in the parent table for the defined variable. It is unique so that only one object gets associated with another.

**What is the use of the Hibernate Session merge() call?**

We can use the Hibernate Session merge() to update existing values; however, this method creates a copy from the passed entity object and returns it. The returned object is part of the persistent context and tracked for any changes; the given object is not tracked. For example, a program, read Hibernate merge.

**What will happen if a user doesn’t have no-args constructor in Entity bean?**

Hibernate uses Reflection API to create an instance of Entity beans while calling get() or load() method. The method Class.newInstance() is used for this, and it requires no-args constructor. So if you won’t have no-args constructor in entity beans, hibernate will fail to instantiate it, and you will get HibernateException.

**What are the collection types in Hibernate?**

The Five collection types in hibernate used for one-to-many relationship mappings are as follows.

* Bag
* Set
* List
* Array
* Map

**How to implement relationships in hibernate?**

We can quickly implement relationships like One to One Mapping, One to Many Mapping, and Many to Many Mappingusing JPA annotations as well as XML based configurations.

**How to use JNDI DataSource with the Hibernate framework?**

It’s always useful to allow a servlet container to manage the connection pool as the user so that we define JNDI resource for DataSource, and we can use it in the web application. It’s straightforward to use in Hibernate, all we need is to remove all the database-specific properties and use the below feature to provide the JNDI DataSource name.

**Example**

<property

### What is a One-to-One association in Hibernate?

* A one to one association is similar to many to one association, and only one difference is that it must be set as a unique one. We can use many to one element to define one to one association.
* A name attribute must set in the parent class, and the column attribute is used to set a column name in the parent table for the defined variable. It is unique so that only one object gets associated with another.

### What is the use of the Hibernate Session merge() call?

We can use the Hibernate Session merge() to update existing values; however, this method creates a copy from the passed entity object and returns it. The returned object is part of the persistent context and tracked for any changes; the given object is not tracked. For example, a program, read Hibernate merge.

### What will happen if a user doesn’t have no-args constructor in Entity bean?

Hibernate uses Reflection API to create an instance of Entity beans while calling get() or load() method. The method Class.newInstance() is used for this, and it requires no-args constructor. So if you won’t have no-args constructor in entity beans, hibernate will fail to instantiate it, and you will get HibernateException.

### What are the collection types in Hibernate?

The Five collection types in hibernate used for one-to-many relationship mappings are as follows.

* Bag
* Set
* List
* Array
* Map

### How to implement relationships in hibernate?

We can quickly implement relationships like One to One Mapping, One to Many Mapping, and Many to Many Mappingusing JPA annotations as well as XML based configurations.

### How to use JNDI DataSource with the Hibernate framework?

It’s always useful to allow a servlet container to manage the connection pool as the user so that we define JNDI resource for DataSource, and we can use it in the web application. It’s straightforward to use in Hibernate, all we need is to remove all the database-specific properties and use the below feature to provide the JNDI DataSource name.