1. What is hash code? Can we override and return same hashcode for two different objects? What is the contract between hashcode and equals() method.
2. Difference between HashMap andHashSet?
3. What data structure would you recommend for ordered and sorted data?
4. Explain exception hierarchy
5. Explain life cycle of thread.
6. Explain about join in SQL.Write a query for outer join.
7. Tell about procedures,Sequences and triggers.
8. Do you know UNIX? Tell some LINUX commands that you know.
9. What you know about jQuery?
10. What is garbage collection? What happens if you say System.gc()?
11. Can we have try statement without catch? IF try statement contains return will the finally block be executed? What happens if there is an exception inside finally block?
12. Can we define enum inside a method?
13. What is polymorphism? What are the types of polymorphism and explain.
14. Take login page example andtell what are the files and configurations needed and explain the life cycle.
15. Query to retrieve maximum salary in each department in a company.
16. What is Generics
17. A problem statement was given –Three colors of ball RGB each different size. Display the ball in descending order of size. Use any logic in java.
18. What does transient means? What JVM does on seeing transient?
19. Explain autoboxing and unboxing.
20. How can you synchronize a list?
21. Some questions about SONAR code quality management. What do you mean by quality code?
22. Explain webservice. What are the types of web service? WSDL is SOAP or Restfull?
23. What are the new features in JSF2.0?
24. What is the tag for ajax support in JSF?
25. Can we override a constructor?
26. What are the views in ClearCase?
27. What are the methods to create a Thread?
28. Why spring framework is very famous?
29. Advantages of JSF framework?
30. Difference between JDK, JVM and JRE.
31. Difference between HashMap and LinkedHashMap?
32. How will you retrieve values and keys from HashMap?
33. Difference between List and Set.

**Java/J2EE Developer** . The interview questions asked are the following -  
  
**Round 1(Written)**

1. Write a java program to implement Quick sort  
2. Write a java program to implement Binary search  
3. Write a java program to find a string in a file  
4. Some question on connection pool ( do not remember exactly)

**Round 2 - ( F2F)**  
Few of the questions that I remember are -   
1. Interchange two strings without using a temporary variable.  
2. Given a list of 2 Arraylists , find the common objects and uncommon objects between them.  
3. Questions on collections ( DIfferent types of collections , when to use which collection , significance of each )  
4. Hash Code  & equals implementation and its significance   
5. Difference between String , StringBuffer,String Builder ( In which version was string builder introduced in java )  
6. Concurrent Package in java ( More on concurrent hashmap and its comparison with HashTable)  
7. Thread concepts (synchronisaton ) and significance and overhead of immutability related to threads.

[**What is the difference between a JVM and JIT compiler (just in time compiler) in Java?**](https://www.quora.com/What-is-the-difference-between-a-JVM-and-JIT-compiler-just-in-time-compiler-in-Java)

JVM (“Java Virtual Machine”) has two meanings:

1. It is an abstract instruction set designed to run Java Programs. This instruction set defines a relatively straightforward push-down stack machine to which Java source programs are compiled; these instructions are stored in “.class” files.
2. It is a program that runs on a real computer, than can execute the abstract JVM instructions. Such a JVM includes an “interpreter” for the individual JVM instructions, but also includes all the supporting machinery required to support the execution of Java code, including arithmetic, function calls, storage allocation, garbage collection, thread scheduling, class file loaders, file I/O and other access to the local operating system as needed to run complex Java applications.

Older JVMs literally interpreted the JVM instructions one-by-one at runtime. This is easy to implement which is why this was done. But like any interpreter, such run-time interpretation produces program execution times that are typically an order of magnitude slower than native compiled machine code.

A “JIT compiler” (“JITter”) is a feature of most modern JVMs that compiles chunks of JVM instructions to native machine code as new chunks, or previously encountered chunks of JVM code are encountered during execution. One could use an offline Java-to-native-code compiler (some actually exist) to do this to provide traditional batch-compiling purposes but that isn’t the way the Java world went.

JITters have one advantage over batch compilers: they can compile code that matches what happens at runtime, and can thus help optimize the dynamic properties of Java code such as the dynamic dispatch that occurs in overloaded Java method calls. They have a disadvantage: they don’t see the program at large scale, and generally cannot do what good batch compilers do well: global optimizations. The tradeoff, in practice, means Java programs that are JIT-compiled running pretty well, but not as fast as C or C++ programs that are compiled by traditional methods.

**Why is String immutable in Java ?**

1. String Pool - When a string is created and if it exists in the pool, the reference of the existing string will be returned instead of creating a new object. If string is not immutable, changing the string with one reference will lead to the wrong value for the other references.  
     
   Example -  
     
   String str1 = "String1";  
   String str2 = "String1"; // It doesn't create a new String and rather reuses the string literal from pool
2. To Cache its Hashcode - If string is not immutable, One can change its hashcode and hence it's not fit to be cached.
3. Security - String is widely used as parameter for many java classes, e.g. network connection, opening files, etc. Making it mutable might possess threats due to interception by the other code segment.

**Explain the scenerios to choose between String , StringBuilder and StringBuffer ?**

**or  
  
What is the difference between String , StringBuilder and StringBuffer ?**

|  |  |  |  |
| --- | --- | --- | --- |
| If the Object value will not change, use String Class because a String object is immutable.  If the Object value can change and will only be modified from a single thread, use StringBuilder because StringBuilder is unsynchronized (means faster).  If the Object value may change, and can be modified by multiple threads, use a StringBuffer because StringBuffer is thread safe (synchronized).  **What is the difference between final, finally and finalize() ?**  final - constant variable, objects cannot be de-referenced, restricting method overriding, restricting class sub classing.  finally - handles exception. The finally block is optional and provides a mechanism to clean up regardless of what happens within the try block. Use the finally block to close files or to release other system resources like database connections, statements etc.  finalize() - method helps in garbage collection. A method that is invoked before an object is discarded by the garbage collector, allowing it to clean up its state.  **How can we run a java program without making any object?**   |  | | --- | | By putting code within static method. With Java 6 and earlier versions, even static block can be used. | |  | |  |

**What is the difference between ArrayList and LinkedList ?**

Underlying data structure for ArrayList is Array whereas LinkedList is the linked list and hence have following differences -  
  
1. ArrayList needs continuous memory locations and hence need to be moved to a bigger space if new elements are to be added to a filled array which is not required for LinkedList.  
  
2. Removal and Insertion at specific place in ArrayList requires moving all elements and hence leads to O(n) insertions and removal whereas its constant O(1) for LinkedList.  
  
3. Random access using index in ArrayList is faster than LinkedList which requires traversing the complete list through references.  
  
4. Though Linear Search takes Similar Time for both, Binary Search using LinkedList requires creating new Model called Binary Search Tree which is slower but offers constant time insertion and deletion.  
  
5. For a set of integers you want to sort using quicksort, it's probably faster to use an array; for a set of large structures you want to sort using selection sort, a linked list will be faster.

**What is the use of hashcode in Java ?**

Hashcode is used for bucketing in Hash implementations like HashMap, HashTable, HashSet etc. The value received from hashcode() is used as bucket number for storing elements. This bucket number is the address of the element inside the set/map. when you do contains() then it will take the hashcode of the element, then look for the bucket where hashcode points to and if more than 1 element is found in the same bucket (multiple objects can have the same hashcode) then it uses the equals() method to evaluate if object are equal, and then decide if contain() is true or false, or decide if element could be added in the set or not.

**Differences between abstract class and interface ?**

|  |
| --- |
| Abstract classes can have both abstract methods ( method declarations ) as well as concrete methods ( inherited to the derived classes ) whereas Interfaces can only have abstract methods ( method declarations ).  A class can extend single abstract class whereas it can implement multiple interfaces. |
|  |
|  |

**Can you give a real world example of Encapsulation and Abstraction ?**

|  |
| --- |
| Car Engine is an example of encapsulation and abstraction. You ignite the car using an interface called starter and least bothered about how the tire actually moves (This is abstraction). The engine encapsulates the complete process to itself only and doesn't allow you to start the other components like the radiator etc ( this is excapsulation ) |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **What is a final method ?** |  |  | |
| Ans. Its a method which cannot be overridden. Compiler throws an error if we try to override a method which has been declared final in the parent class. |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Explain Thread States ?** |  |  | |
| Ans. Runnable - waiting for its turn to be picked for execution by the thread schedular based on thread priorities.  Running - The processor is actively executing the thread code. It runs until it becomes blocked, or voluntarily gives up its turn.   Waiting: A thread is in a blocked state while it waits for some external processing such as file I/O to finish.  Sleeping - Java threads are forcibly put to sleep (suspended) with Thread.sleep. they can resume using Thread.resume method.  Blocked on I/O - Will move to runnable after I/O condition like reading bytes of data etc changes.  Blocked on synchronization - Will move to Runnable when a lock is acquired.  Dead - The thread is finished working. |

**Explain multithreading in Java ?**

|  |
| --- |
| Multithreading provides better interaction with the user by distribution of task  2. Threads in Java appear to run concurrently, so it provides simulation for simultaneous activities.The processor runs each thread for a short time and switches among the threads to simulate sim-ultaneous execution (context-switching) and it make appears that each thread has its own processor.By using this feature, users can make it appear as if multiple tasks are occurring simultaneously when, in fact, each is running for only a brief time before the context is switched to the next thread.  3. We can do other things while waiting for slow I/O operations.In the java.iopackage, the class InputStreamhas a method, read(), that blocks until a byte is read from the stream or until an IOExceptionis thrown. The thread that executes this method cannot do anything elsewhile awaiting the arrival of another byte on the stream. |
|  |
|  |

**What is the difference between List, Set and Map ?**List - Members are stored in sequence in memory and can be accessed through index.   
Set - There is no relevance of sequence and index. Sets doesn't contain duplicates whereas multiset can have duplicates.   
Map - Contains Key , Value pairs.

**What is the use of Transient Keyword ?**

|  |
| --- |
| It in Java is used to indicate that a field should not be serialized. |
|  |
|  |

**What is an Immutable Object ?**

Object that can't be changed after instantiation.

**What is an immutable class ?**

|  |
| --- |
| Class using which only immutable (objects that cannot be changed after initialization) objects can be created. |
|  |

**Explain Autoboxing ?**

Autoboxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes

**What is reflection ?**

It is the process of examining / modifying the behaviour of an object at runtime.

**Difference between Checked and Unchecked exceptions ?**

Checked exceptions are the exceptions for which compiler throws an errors if they are not checked

whereas unchecked exceptions are caught during run time only and hence can't be checked.

**Checked Exceptions**

1. ClassNotFoundException: Class not found.
2. CloneNotSupportedException: Attempt to clone an object that does not implement the Cloneable interface.
3. IllegalAccessException: Access to a class is denied.
4. InstantiationException: Attempt to create an object of an abstract class or interface.
5. InterruptedException: One thread has been interrupted by another thread.
6. NoSuchFieldException: A requested field does not exist.
7. NoSuchMethodException: A requested method does not exist.
8. ReflectiveOperationException - Superclass of reflection -related exceptions(Added by JDK 7.)

## Runtime - Unchecked Exceptions

1. ArithmeticException: Arithmetic error, such as divide-by-zero.
2. ArrayIndexOutOfBoundsException: Array index is out-of-bounds.
3. ArrayStoreException: Assignment to an array element of an incompatible type.
4. ClassCastException: Invalid cast.
5. IllegalArgumentException: Illegal argument used to invoke a method.
6. IllegalStateException: Environment or application is in incorrect state.
7. IllegalThreadStateException: Requested operation not compatible with current thread state.
8. IndexOutOfBoundsException: Some type of index is out-of-bounds.
9. NegativeArraySizeException: Array created with a negative size.
10. NullPointerException: Invalid use of a null reference.
11. NumberFormatException: Invalid conversion of a string to a numeric format.
12. SecurityException: Attempt to violate security.
13. StringIndexOutOfBounds: Attempt to index outside the bounds of a string.
14. TypeNotPresentException: Type not found. (Added by J2SE 5.)
15. UnsupportedOperationException: An unsupported operation was encountered.

**What is an Iterator?**

|  |
| --- |
| Iterator is an interface that provides methods to iterate over any Collection. |
|  |
|  |

**What is a class loader ? What are the different class loaders used by JVM ?**

Part of JVM which is used to load classes and interfaces.  
  
Bootstrap , Extension and System are the class loaders used by JVM.

**What is PermGen or Permanent Generation ?**

|  |
| --- |
| The memory pool containing all the reflective data of the java virtual machine itself, such as class and method objects. With Java VMs that use class data sharing, this generation is divided into read-only and read-write areas. The Permanent generation contains metadata required by the JVM to describe the classes and methods used in the application. The permanent generation is populated by the JVM at runtime based on classes in use by the application. In addition, Java SE library classes and methods may be stored here. |
|  |
|  |

**Explain OOPs**

OOPs or Object Oriented Programming is a Programming model which is organized around Objects instead of processes. Instead of a process calling series of processes, this model stresses on communication between objects. Objects that all self sustained, provide security by encapsulating it's members and providing abstracted interfaces over the functions it performs. OOP's facilitate the following features  
  
1. Inheritance for Code Reuse  
2. Abstraction for modularity, maintenance and agility  
3. Encapsulation for security and protected  
4. Polymorphism for flexibility and interfacing

**What is deployment descriptor ?**

Deployment Descriptor which is usually web.xml is used to specify the classes, resources and configuration of the application and how the web server uses them to serve web requests.This file is usually added to WEB-INF folder and contains following   
  
\* Servlet entries and url mapping  
\* Plugins  
\* Some info regarding authentication / filters  
\* Landing Page  
\* Event Handlers

**What is Dependency Injection or IOC ( Inversion of Control ) ?**

 It is a Design Pattern that facilitates loose coupling by sending the dependency information

(object references of dependent object ) while building the state of the object. Objects are designed in a manner where they receive instances of the objects from other pieces of code, instead of constructing them internally and hence provide better flexibility.

**What are the contents of Hibernate configuration file ( hibernate.cfg.xml ) ?**

|  |
| --- |
| HBM Files ( Mapping ) DB Connection ( DB Connection String , User Name , Password , Pool Size ) SQL Dialect ( SQL variant to be generated ) Show SQL ( Show / No show SQL on Console ) Auto Commit ( True / False ) |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Difference between first level and second level cache in hibernate ?** |  | |  |  | | --- | --- | |  |  | | |
| Ans. 1. First level cache is enabled by default whereas Second level cache needs to be enabled explicitly.  2. First level Cache came with Hibernate 1.0 whereas Second level cache came with Hibernate 3.0.  3. First level Cache is Session specific whereas Second level cache is shared by sessions that is why First level cache is considered local and second level cache is considered global. |

**Explain Flow of Spring MVC ?**

The DispatcherServlet configured in web.xml file receives the request.  
  
The DispatcherServlet finds the appropriate Controller with the help of HandlerMapping and then invokes associated Controller.  
  
Then the Controller executes the logic business logic and then returns ModeAndView object to the DispatcherServlet.  
  
The DispatcherServlet determines the view from the ModelAndView object.  
  
Then the DispatcherServlet passes the model object to the View.  
  
The View is rendered and the Dispatcher Servlet sends the output to the Servlet container.   
  
Finally Servlet Container sends the result back to the user.

**What are RESTful Web Services ?**

|  |
| --- |
| REST or Representational State Transfer is a flexible architecture style for creating web services that recommends the following guidelines -   1. http for client server communication,  2. XML / JSON as formatting language,  3. Simple URI as address for the services and,  4. State less communication. |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **What are Inner , Outer , Left and Right Joins in SQL ?** | [Database](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Database) | 2017-01-09 13:07:55   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. Inner join is the intersection of two tables on the condition defined by the where clause i.e will get records from both tables matched by a column.  Outer join is the union of two tables i.e will get all records from both tables and will put null in the columns where related records are not present.  Left Outer join is the left union of two tables i.e all records from the table on the left and values from the right table for related records else null for the columns from right table.   Right Outer join is the right union of two tables i.e all records from the table on the right and values from the left table for related records else null for the columns from left table. |

**What is a self Join and give an example of a self Join ?**

|  |
| --- |
| When a Table Join itself , it's a Self Join. For example - we like to know the pair of department names where first dept has lesser employees than the later.  Select D1.name , D2.name from Dept D1, Dept D2 where D1.employee\_count < D2.employee\_count |
|  |

J2EE

**What are the components of J2EE applications?**

* Client-tier components. Run on the client machine.
* Web tier components. Run on the J2EE server.
* Business tier components. Run on the J2EE server.
* Enterprise information system software (EIS software).Runs on the **EIS** server.

**What are the J2EE client types?**

* Applets
* Application clients
* Java Web Start enabled clients, by **Java Web Start technology**.
* Wireless clients, based on **Mobile Information Device Profile (MIDP)** technology.

**What is a J2EE container?**

The interface between a component and the low-level platform with specific functionality that support the component is called a **container**. Application server maintains control and provides relevant services through an interface or framework calls as a **container**.

**What are defined as web components?**

**Java Servlets** and Java server pages technology components are identified as web components. Servlets dynamically receive requests and make responses. JSP server pages also behave as **Java Servlets** but facilitate to create more static contents.

**Define JSF?**

**JSF** stands for Java Server Faces. It is the user interface (UI) designing framework for Java Web Applications developments. There are set of reusable UI components associated with JSF. Also, JSF based on Model-View-Controller (MVC) design concepts and patterns. The automated data saving process from form to server and display at client side is also handling by JSF.

**What is Hashtable?**

Hash table is a Collection Synchronized objects. It allows Null values but not duplicate values. Hash table is like a HashMap.

**Define Hibernate?**

Hibernate is an open source object-relational mapping and query service which facilitate to write **Hibernate Query Language (HQL)** scripts instead of **Structured Query Language (SQL)** scripts. It is faster and easy than writing native **SQL**. Hibernate has more powerful object oriented contents like associations, inheritance, and polymorphism. Also, Hibernate has a powerful composition and collections. Hibernate allows making queries using Java base approach.

**What are the identified limitation of hibernate?**

* **Slower in action –** In execution of HQL queries take more time than it executes directly.
* Only composite keys support available and it prevents advanced query options.
* No shared value type references available.

**What the identified advantages are of hibernate?**

* Database and vendor independence application.
* Standard Object-relational mapping support.
* Domain object mapping for a relational database.
* Better performance than Java Database Connectivity.
* Java Persistence **API**based applications.

**Describe ORM?.**

**Object-Relational mapping (ORM)** can describe as follows.

The mapped objects in a Java class to the tables of the relational database using metadata which describes the database and object mapping. The working method is to transform data from one representation to another.

**What are the advantages of Object-Relational mapping (ORM)?**

* **Productivity –** Reduce the time for data access coding with help of automatic code creation base on the defined data model.
* **Maintainability –** All code generated from **ORM** are well tested. Only the developer need to create the correct functionality
* **Performance –** The code generated from **ORM** completely manages the data access needs of the application. No need to create any data access code to create and also the code is optimized for speed up the data access process.
* **Vendor independence –** The code generated from **ORM** is not depending on the vendor. This is to increase the portability of the application.

**What is the use of method save() in hibernate?**

 In hibernate this method is used to stores an object into the database. There is a check for duplicate records before inserting it.

**What is the use of method saveorupdate()?**

 In hibernate this method is used to update an object using identifier. When the value for the identifier is NULL then the method direct to call save().

**What is the difference between load() and get()?**

When the object not available in either cache or database, load() thrown an exception. No null return from load().

When the object not available in either cache or database, get() returns null.

**What is mean by connection pooling?**

Simply connection pooling is a mechanism to re-use the existing connections. The **pooling** mechanism maintains a number of already created object connections and when there is a demand the mechanism directly use existing connection without creating a new one.

**Define the Collection types in Hibernate?**

One-to-many reference is defined as a collection. There are five main collection types associated with J2EE.They are

Set type, List type, Array type, Map type, Bag type

**Define thin client?**

A program interface that does not have any operations like database queries, complex business rules or any connection to the third-party application is called a thin client.

**Describe the file types \*.ear, \* .jar and \*.war?**

* **\*.jar files –** Property file contains libraries, resources and accessories are included with the \*.jar file extension.
* **\*.war files –** The files that need to development of web application (HTML, java scripts, JSP) included with a \*.war file extension.
* **\*.ear files –** The files for Enterprise Java Beans modules for the application is save as \*.ear files.

**What spring is in related to J2EE?**

**Spring** is an open source application that reduces the complexity of enterprise application development. Spring is based on an inversion of control or dependency injection design patterns.

**What are the advantages in the use of spring for application development?**

* **Plain Old Java Object (POJO)** based development facilitates to re-use existing components.
* Possible to reduce development cost by improving the productivity of the application development.
* Improve the testability of application with dependency injection.
* Improve maintainability with reduce code coupling.
* No need to have an application server and works on enterprise service.

**Discuss the benefit of Spring Framework?**

* Possibility to organize middle tier objects in an efficient way.
* Easy initialization for properties.
* Easily testable components.
* Lightweight container.
* Possibilities to use configure management service of spring in any runtime environment with whatever architectural layer.

**Describe servlet?**

Server side component that provides a powerful mechanism to create server side programs is called a **servlet**. There are servlets available with a design for various protocols. Servlet is also server and platform independent. Most commonly use protocol for the servlet is **Hypertext Transfer Protocol (HTTP)**.Also, a servlet is a pure java object.

**Describe Secure Socket Layer (SSL)?**

The technology that use to communicate between the web server and the web browser is called **Secure Socket Layer (SSL)**.More especially SSL is a protocol that describes how algorithms to be used in encryption. The technology established an encrypted link between two parties and this link is allowed to secure transmission of sensitive information such as login credentials, credit/debit card information and social security numbers.

**What id URL?**

**URL** stands for Uniform Resource Locator and it is the textual reference writing standard to an arbitrary piece of data in the **World Wide Web (WWW)**.The general structure of URL is as follows.

***protocol://host/local info***

* ***protocol* ->** Protocol for fetching the object ( eg – HTTP, FTP)
* ***host* ->** Internet name of the targeted host.
* ***local info* ->** String passed to the protocol handler on the remote host. In many cases, it is a file name with extension.

**What is URN?**

**URN** stands for Uniform resource name. It is the unique identifier that identified an entity. But the information of where the entity is located is not available.

**What are the phases of a servlet life cycle?**

* Loading of the servlet class.
* Instantiation of Servlet.
* Execution of init method.
* Request handling phase. In this phase service methods will call.
* Removal from service phase. In this phase destroy method will call.

**Describe the phases of servlet lifecycle?**

* **Class loading phase –** Web container loads the servlet class file (\*.class).
* **Instantiation phase –** By calling default no-arg constructor the servlet class gets Instantiated.
* **Initialize phase –** The method Init () called in this phase in only one time of the lifetime of a servlet. Servlet configuration is assigned to the servlet.
* **Request Handling phase –** In this phase, only servlets spends most of the time. Servlet provides the services to various requests by calling Service ().
* **Removal phase –** The destroy () function calls before servlet destruction. Garbage collection occurs later.

**What are the difference types of JSP tags?**

***There are 4 different types of tags associated with JSP.They are mentioned below***

* Directives
* Declarations
* Scriplets
* Expressions

**What are the two types of comments supported by JSP?**

***There are two types of comments are supported by JSP.***

* HTML comment.

[HTML comment](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/HTML-comment.jpg)

* JSP comment.

[JSP comment](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/JSP-comment.jpg)

**What is called JSP directive?**

**JSP** directive is the mechanism to provide Metadata information to web container about JSP file. In the translation and compilation phases of the **JSP** life cycle, these Metadata use by the web container.

**What are the different types of JSP directive?**

* Page directive
* Include directive
* Taglib directive

**What is EJB?**

**EJB** stands for Enterprise Java Beans. It is the server side components that executes in EJB container and encapsulates the business logic for the enterprise application.

**What are the system services of EJB container?**

***EJB Container provides following system services.***

* Persistence
* Security
* Transaction
* Connection pooling
* Component lifecycle management
* Threading

**What are the design principles for EJB?**

* Behavior of the **EJB** application is specified by interfaces
* EJB applications are loosely coupled and tired.
* Implementation is hidden from the client side.
* The **EJB** container supports the application developer.
* The API to the application is in session tier.
* The API to the data sources is in entity tier.

**What are the advantages of EJB components?**

* There is a support for the integration of components from different vendors.
* Possibility authored **EJB** components detailed knowledge of the environment.
* Possibility to assembled applications from separate components source.
* Interaction with its clients is entirely specified in terms of java interfaces.
* Portability support.
* Does not maintain resources

**What are the Basic and subtypes of Enterprise Java Beans (EJB)?**

* **Session Beans**
  + Stateful session beans
  + Stateless session beans
* **Entity Beans**
  + Bean Managed Persistence (BMP)
  + Container Managed Persistence (CMP)
  + Message Driven Beans

**What Is J2ee?**

J2EE is an environment for developing and deploying enterprise applications. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multitiered, web-based applications.

**What Is The J2ee Module?**

A J2EE module consists of one or more J2EE components for the same container type and one component deployment descriptor of that type.

**What Are The Components Of J2ee Application?**

A J2EE component is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and communicates with other components. The J2EE specification defines the following J2EE components:

Application clients and applets are client components.

Java Servlet and JavaServer PagesTM (JSPTM) technology components are web components.

Enterprise JavaBeansTM (EJBTM) components (enterprise beans) are business components.

Resource adapter components provided by EIS and tool vendors.

**What Are The Four Types Of J2ee Modules?**

. Application client module  
2. Web module  
3. Enterprise JavaBeans module  
4. Resource adapter module.

**What Does Application Client Module Contain?**

The application client module contains:  
--class files,  
--an application client deployment descriptoor.  
Application client modules are packaged as JAR files with a .jar extension.

**What Does Web Module Contain?**

JSP files,  
--class files for servlets,  
--GIF and HTML files, and  
--a Web deployment descriptor.  
Web modules are packaged as JAR files with a .war (Web ARchive) extension.

**What Are The Differences Between Ear, Jar And War Files? Under What Circumstances Should We Use Each One?**

There are no structural differences between the files; they are all archived using zip-jar compression.  
However, they are intended for different purposes.  
--Jar files (files with a .jar extension) are intended to hold generic libraries of Java classes, resources, auxiliary files, etc.  
--War files (files with a .war extension) arre intended to contain complete Web applications. In this context, a Web application is defined as a single group of files, classes, resources, .jar files that can be packaged and accessed as one servlet context.  
--Ear files (files with a .ear extension) arre intended to contain complete enterprise applications. In this context, an enterprise application is defined as a collection of .jar files, resources, classes, and multiple Web applications.  
Each type of file (.jar, .war, .ear) is processed uniquely by application servers, servlet containers, EJB containers, etc.

**What Is The Difference Between Session Bean And Entity Bean?one?**

**Answer :**

The Session bean and Entity bean are two main parts of EJB container.  
***Session Bean***  
--represents a workflow on behalf of a client.  
--one-to-one logical mapping to a client.  
--created and destroyed by a client.  
--not permanent objects.  
--lives its EJB container(generally) does noot survive system shut down.  
--two types: stateless and stateful beans Entity Bean.  
--represents persistent data and behavior off this data.  
--can be shared among multiple clients.  
--persists across multiple invocations.  
--findable permanent objects.  
--outlives its EJB container, survives systeem shutdown.  
--two types: container managed persistence(CCMP) and bean managed persistence(BMP).

**What Is Authorization?**

he process by which access to a method or resource is determined. Authorization depends on the determination of whether the principal associated with a request through authentication is in a given security role. A security role is a logical grouping of users defined by the person who assembles the application. A deployer maps security roles to security identities. Security identities may be principals or groups in the operational environment.

**What Is Authorization Constraint?**

An authorization rule that determines who is permitted to access a Web resource collection.

**What Is Basic Authentication?**

An authentication mechanism in which a Web server authenticates an entity via a user name and password obtained using the Web application's built-in authentication mechanism.

**What Is Bean-managed Persistence?**

The mechanism whereby data transfer between an entity bean's variables and a resource manager is managed by the entity bean.

**What Is Bean-managed Transaction?**

A transaction whose boundaries are defined by an enterprise bean.

**What Is Binding (xml)?**

Generating the code needed to process a well-defined portion of XML data.

**What Is Business Logic?**

The code that implements the functionality of an application. In the Enterprise JavaBeans architecture, this logic is implemented by the methods of an enterprise bean.

**What Is Business Method?**

A method of an enterprise bean that implements the business logic or rules of an application.

**What Is Callback Methods?**

Component methods called by the container to notify the component of important events in its life cycle.

**What Is Cascade Delete?**

A deletion that triggers another deletion. A cascade delete can be specified for an entity bean that has container-managed persistence.

**What Is Cdata?**

A predefined XML tag for character data that means "don't interpret these characters," as opposed to parsed character data (PCDATA), in which the normal rules of XML syntax apply. CDATA sections are typically used to show examples of XML syntax.

**What Is Container?**

An entity that provides life-cycle management, security, deployment, and runtime services to J2EE components. Each type of container (EJB, Web, JSP, servlet, applet, and application client) also provides component-specific services.

**What Is Context Root?**

A name that gets mapped to the document root of a Web application.

**What Is Cts?**

Compatibility test suite. A suite of compatibility tests for verifying that a J2EE product complies with the J2EE platform specification.

**What Is Document Object Model?**

An API for accessing and manipulating XML documents as tree structures. DOM provides platform-neutral, language-neutral interfaces that enables programs and scripts to dynamically access and modify content and structure in XML documents.

**What Is Enterprise Bean?**

A J2EE component that implements a business task or business entity and is hosted by an EJB container; either an entity bean, a session bean, or a message-driven bean.

**What Is Filter?**

An object that can transform the header or content (or both) of a request or response. Filters differ from Web components in that they usually do not themselves create responses but rather modify or adapt the requests for a resource, and modify or adapt responses from a resource. A filter should not have any dependencies on a Web resource for which it is acting as a filter so that it can be composable with more than one type of Web resource.

**What Is Filter Chain?**

A concatenation of XSLT transformations in which the output of one transformation becomes the input of the next.

**What Is Form-based Authentication?**

provides an application-specific form for logging in. This form of authentication uses Base64 encoding and can expose user names and passwords unless all connections are over SSL.

**What Is Java Transaction Api (jta)?**

An API that allows applications and J2EE servers to access transactions.

**What Is Javaserver Pages Standard Tag Library (jstl)?**

A tag library that encapsulates core functionality common to many JSP applications. JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization and locale-specific formatting tags, SQL tags, and functions.

**What Is Java Naming And Directory Service?**

The JNDI provides naming and directory functionality. It provides applications with methods for performing standard directory operations, such as associating attributes with objects and searching for objects using their attributes. Using JNDI, a J2EE application can store and retrieve any type of named Java object. Because JNDI is independent of any specific implementations, applications can use JNDI to access multiple naming and directory services, including existing naming and directory services such as LDAP, NDS, DNS, and NIS.

**What Is The Diffrence Between Java Mail And Jms Queue?**

**Java Mail** - API siting on top of e-mail protocols like SMTP, POP, IMAP - essentially same stuff e-mail clients like MS outlook use .. hence make sense if at least on one side of conversation we have human.   
  
**JMS Queue** - is asynchronous point-to-point communication between systems

**How Do You Optimize Servlets?**

1. Use init() method to cache static data.  
   • Use StringBuffer rather than using + operator when you concatenate multiple strings.  
   • Use print() method rather than println() method.  
   • Use ServletOutputStream rather than PrintWriter to send binary data.  
   • Initialize the PrintWriter with proper size.  
   • Flush the data partly.  
   • Minimize code in the synchronized block.  
   • Set the content length  
   • Release resources in destroy() method.  
   • Implement getLastModified() method to use browser cache and server cache  
   • Use application server caching facility  
   • Use Mixed session mechanisms such as HttpSession with hidden fields  
   • Remove HttpSession objects explicitly in your program whenever you finish the task.  
   • Reduce session time out value as much as possible.  
   • Use 'transient' variables to reduce serialization overhead if your HttpSession tracking mechanism uses serialization process.  
   • Disable servlet auto reloading feature.  
   • Use thread pool for your servlet engine and define the size as per application requirement.

**How Do You Optimize A Jsp Page?**

• Use jspInit() method to cache static data  
• Use StringBuffer rather than using + operator when you concatenate multiple strings  
• Use print() method rather than println() method  
• Use ServletOutputStream instead of JSPWriter to send binary data  
• Initialize the 'out' object (implicit object) with proper size in the page directive.  
• Flush the data partly.  
• Minimize code in the synchronized block.  
• Set the content length.  
• Release resources in jspDestroy() method.  
• Give 'false' value to the session in the page directive to avoid session object creation.  
• Use include directive instead of include action when you want to include the child page content in the translation phase.  
• Avoid giving unnecessary scope in the 'useBean' action.  
• Do not use custom tags if you do not have reusability.  
• Use application server caching facility.  
• Use Mixed session mechanisms such as 'session' with hidden fields.  
• Use 'session' and 'application' as cache.  
• Use caching tags provided by different organizations like openSymphony.com.  
• Remove 'session' objects explicitly in your program whenever you finish the task.  
• Reduce session time out value as much as possible.  
• Use 'transient' variables to reduce serialization overhead if your session tracking mechanism uses serialization process.  
• Disable JSP auto reloading feature.

**How Do You Create Multiple Virtual Hosts?**

If you want tomcat to accept requests for different hosts e.g., www.myhostname.com then you must   
  
0. create ${catalina.home}/www/appBase , ${catalina.home}/www/deploy, and ${catalina.home} /conf /Catalina /www.myhostname.com   
  
1. add a host entry in the server.xml file   
  <Host appBase="www/appBase" name="www.myhostname.com"/>  
  
2. Create the the following file under conf/Catalina/www.myhostname.com/ROOT.xml   
<?xml version="1.0" encoding="UTF-8"?>  
<Context  
    path="/"  
    docBase="www/deploy/mywebapp.war"  
    reloadable="true" antiJARLocking="true">  
</Context>  
Add any parameters specific to this hosts webapp to this context file   
  
3. put your war file in ${catalina.home}/www/deploy   
  
When tomcat starts, it finds the host entry, then looks for any context files and will start any apps with a context   
  
To add more sites just repeat and rinse, all webapps can share the same war file location and appbase.

**How Will You Load Properties File?**

Use a ResourceBundle. See the Java docs for the specifics of how the ResourceBundle class works. Using this method, the properties file must go into the WEB-INF/classes directory or in a jar file contained in the WEB-INF/lib directory.  
• Another way is to use the method getResourceAsStream() from the ServletContext class. This allows you update the file without having to reload the webapp as required by the first method. Here is an example code snippet, without any error trapping:

// Assuming you are in a Servlet extending HttpServlet   
// This will look for a file called "/more/cowbell.properties" relative  
// to your servlet Root Context  
InputStream is = getServletContext().getResourceAsStream("/more/cowbell.properties");  
Properties p = new Properties();  
p.load(is);  
is.close();

**What Is The Difference Between Using Getsession(true) And Getsession(false) Methods?**

getSession(true) - This method will check whether already a session is existing for the user. If a session is existing, it will return that session object, Otherwise it will create new session object and return taht object.

getSession(false) - This method will check existence of session. If session exists, then it returns the reference of that session object, if not, this methods will return null.