TCP/IP:

TCP – Responsible for making sure that a file sent from one network node to another ends up as a complete file at the destination, even though the file is split into chunks when sent.

IP – Protocol that moves/routes the chunks(Packets) from one host to another on their way to destination.

HTTP – Another network protocol that has web specific features but depends on TCP/IP to get the complete request and response from one place to another.

[**http://en.wikipedia.org/wiki/Sorting\_algorithm**](http://en.wikipedia.org/wiki/Sorting_algorithm)

[**http://www.sorting-algorithms.com/**](http://www.sorting-algorithms.com/)

[**http://tanksoftware.com/tutes/uni/sorting.html#BucketSort**](http://tanksoftware.com/tutes/uni/sorting.html#BucketSort)

**java coded algorithms ---** [**http://www.mycstutorials.com/articles/**](http://www.mycstutorials.com/articles/)

=============================

**What is the difference between a String and StringBuffer.What is the advantage of String Immutability.**

Immutable objects are simply objects whose state (the object's data) cannot change after construction**.**

==========================

Grep and locate commands in UNIX.

==============================

Programme for Fibonacci Series.

0 1 1 2 3 5 8 13 21

public class JavaApplication1 {

public static void main(String[] args) {

// int number\_terms = args[0];

// JavaApplication1.fibonacci(number\_terms);

JavaApplication1.fibonacci(15);

}

public static void fibonacci(int n){

int first\_number=0;

int second\_number=1;

int nterms=n;

for(int i=0;i < nterms; i++){

System.out.println(first\_number);

int new\_firstnumber = second\_number;

second\_number = first\_number + second\_number;

first\_number = new\_firstnumber;

}

}

}

**===============================**

public class X   
{   
 public static void main(String [] args)   
 {  
 try   
 {  
 badMethod();   
 System.out.print("A");   
 }   
 catch (Exception ex)   
 {  
 System.out.print("B");   
 }   
 finally   
 {  
 System.out.print("C");   
 }   
 System.out.print("D");   
}   
 public static void badMethod()   
 {  
 throw new Error(); /\* Line 22 \*/  
 }   
}

**A. ABCD**

**B. Compilation Fails**

**C. C is printed before exiting with an error message.**

**D. BC is printed before exiting with an error message.**

**Answer:** Option **C**

**Explanation:**

Error is thrown but not recognised line(22) because the only catch attempts to catch anException and Exception is not a superclass of Error. Therefore only the code in the finally statement can be run before exiting with a runtime error (Exception in thread "main" java.lang.Error).

**Default value of session attribute in JSP**

The default value of session attribute is true.

====================================

**Difference between Array and ArrayList:**

1. An array needs to know its size when it is created.

2. To put an object in an array you must assign it to a specific location.

3. Arrays use array syntax that is not used any where else in java.

4. ArrayLists in java 5 are parameterized.

=============================

### Why Wait And Notify Methods Are In Object Class Not In Thread?

It was a design decision.

Java’s concurrency model needs “locks” and it was decided that each object was to be associated with a lock.

The wait and notify methods are not associated with a thread but with a lock, so the decision of coupling locks and objects meant that each object should have wait and notify methods that operate on that object’s lock.

**Explanation and Example**

A gas station has a single toilet, the key for which is kept at the service desk. The toilet is a shared resource for passing motorists. To use this shared resource the prospective user must acquire a key to the lock on the toilet. The user goes to the service desk and acquires the key, opens the door, locks it from the inside and uses the facilities.

Meanwhile, if a second prospective user arrives at the gas station he finds the toilet locked and therefore unavailable to him. He goes to the service desk but the key is not there because it is in the hands of the current user. When the current user finishes, he unlocks the door and returns the key to the service desk. He does not bother about waiting customers. The service desk gives thekey to the waiting customer. If more than one prospective user turns up while the toilet is locked, they must form a queue waiting for the key to the lock. Each thread has no idea who is in the toilet.

Obviously in applying this analogy to Java, a Java thread is a user and the toilet is a block of code which the thread wishes to execute. Java provides a way to lock the code for a thread which is currently executing it using the synchorinized keywokd, and making other threads that wish to use it wait until the first thread is finished. These other threads are placed in the waiting state. Java is NOT AS FAIR as the service station because there is no queue for waiting threads. Any one of the waiting threads may get the monitor next, regardless of the order they asked for it. The only guarantee is that all threads will get to use the monitored code sooner or later.

Finally the answer to your question: the lock could be the key object or the service desk. None of which is a Thread.

However, these are the objects that currently decide whether the toilet is locked or open. These are the objects that are in a position to notify that the bath room is open (“notify”) or ask people to wait when it is locked wait.

If the threads were designed to give the lock to one another, then one thread might ‘chose’ a ‘friendly’ thread leading to nepotism.

Hence, the wait and notify methods have to be in Object class.

Threads borrow keys from JVM and return to JVM.

NOW what is a lock?

Locks are inbuilt, hidden objects in a class. For static synchorinized methods the class object has a lock and for non-staic classes the objects instances themselves are the lock.

That is why while blocking a piece of code (instead of an entire method) we use synchronized(this){ …;}

If the same thread has to access various piece of code which are mutually exclusive to modifications then simply create two objects and call them lock 1 and lock 2 and use these two locks to synchornize.

example. if within the same code there are two areas that have to be synchronized and if the two areas are mutually exclusive then we SHOULD NOT USE this keyword for locking.

In our analogy, if there is a condom vending machine in the toilet, then if an user who does not want to buy condoms is using the bathroom, then the same key will lock the toilet and the vending machine and if there is a person in the queue who wants to use the toilet only to get a condom, then he is unnecessarily locked. In this case the gas station has to use two rooms and two keys. That way using one will not affect the other.

Need clarifications let me know.

Most importantly wait, notify , notifyall come into picture only when synchonizing a piece of code.

**What is JDK and JVM ?**

Java Virtual Machine (JVM) is an abstract computing machine. Java Runtime Environment (JRE) is an implementation of the JVM. Java Development Kit (JDK) contains JRE along with various development tools like Java libraries, Java source compilers, Java debuggers, bundling and deployment tools.

JVM becomes an instance of JRE at runtime of a java program. It is widely known as a runtime interpreter. The Java virtual machine (JVM) is the cornerstone on top of which the Java technology is built upon. It is the component of the Java technology responsible for its hardware and platform independence. JVM largely helps in the abstraction of inner implementation from the programmers who make use of libraries for their programmes from JDK.



---------------------------------------------------------

-------------------------------------------------

**If you want print something in java without using static blocks or main method.**

class Google

{

static int i=m1();`

public static int m1()

{

System.out.println("Hi......");

return 10;

}

}

---------- Output ----------

Hi......

java.lang.NoSuchMethodError: main

Exception in thread "main"

------------------------------------------

**What is the difference between yielding and sleeping?**

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

-------------------------------------------

**What is synchronization and why is it important?**

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is 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 significant errors.

---------------------

**Can there be an abstract class with no abstract methods in it?**

**Can an Interface be final?**

**Can an Interface have an inner class?**

----------------------------------------------------------

**Explain the javax.servlet.Servlet interface ?**

**init() :**

**public void init(ServletConfig config) throws ServletException**

It is executed only once after the instantiation of servlet to perform initialization

activities. Web Container doesnot place the servlet into the context if the method

throws any exception or if the method execution is not completed in the specified

time.

**service() :**

**public void service(ServletRequest req,ServletResponse res) throws ServletException,**

**IOException**

This method will be executed for every request. The request and response objects

created for this request are eligible for garbage collection once the service method

execution completes succesfully.

**destroy() :**

**public void destroy()**

This method is executed only once just before taking servlet out of service to

perform clean up activities.

----------

**Difference between WebServer and Web Container?**

**WebServer**: A machine which can take in client request over the web(internet) and give back a static response back to the client (static HTML page, jar file etc)

Ex: Apache Web Server

**Web Container**: Servlets/JSPs have the capability to develop dynamic pages (pages whose content/layouts changes based on the request). Servlets/JSPs live within the Web Container (i.e. Web Container manages the Servlets/JSPs). The WebServer communicates with the Web Container to generate dynamic web pages.

Ex: Tomcat

--------------------------------------------------

**What is a servlet ?**

Servlet is server side web component managed by web container to generate dynamic information as response.

---------------------------------------------------------

**Difference between sleep() and wait() ?**

1) wait() is a method of Object class. sleep() is a method of Thread class.

2) sleep() allows the thread to go to sleep state for x milliseconds. When a thread goes into sleep state it doesn’t release the lock. wait() allows thread to release the lock and goes to suspended state. The thread is only active when a notify() or notifyAll() method is called for the same object

---------------------------------------------------

**Singleton Design Pattern:**

Singleton Design pattern is used to control object creation, limiting the number to one but allowing the flexibility to create more objects if the situation arises.To prevent direct instantiation we create a private default constructor,so that other classes can’t create a new Instance.

public class SingletonObject

{

private static SingletonObject ref;

private SingletonObject()

{

}

public static SingletonObject getSingletonObject()

{

if(ref == null)

{

ref = new SingletonObject();

return ref;

}

}

public Object clone() throws CloneNotSupportedException

{

throws new CloneNotSupportedException();  
}

}

----------------------------------------

**What is the difference between the getRequestDispatcher(String path) method of ServletRequest interface and ServletContext interface?**

The difference between ServletRequest.getRequestDispatcher(String path) and ServletContext.getRequestDispatcher(String path) is that the former can accept a relative path as well whereas the latter can accept paths relative to the current context root only.

If the path starts with a '/' in the getRequestDispatcher(String path) of the ServletRequest interface then it's interpreted as being relative to the current context root otherwise it'll be a relative to the request of the calling servlet. Whereas, the path of the getRequestDispatcher(String path) of the ServletContext interface must start with '/' only - being relative to the current context root.

Another difference between the two is that path of the getRequestDispatche(String path) of the ServletRequest interface cannot extend outside the current servlet context whereas getRequestDispatcher(String path) of the ServletContext can use the [getContext](http://java.sun.com/javaee/5/docs/api/javax/servlet/ServletContext.html#getContext%28java.lang.String%29)(String uripath) method to obtain RequestDispatcher for resources in foreign contexts.

Relative Path is relative to the current location and never starts with ‘/’.

Absolute Path represents the path relative to context root. It should start with ‘/’

-----------------------------------------------------------------------------------------

**Polymorphism:**

Polymorphism provides one of the most useful programming techniques of the object-oriented paradigm. *Polymorphism*, which etymologically means "many forms," is the ability to treat an object of any subclass of a base class as if it were an object of the base class. A base class has, therefore, many forms: the base class itself, and any of its subclasses.

If you need to write code that deals with a family of types, the code can ignore type-specific details and just interact with the base type of the family. Even though the code thinks it is sending messages to an object of the base class, the object's class could actually be the base class or any one of its subclasses. This makes your code easier for you to write and easier for others to understand. It also makes your code extensible, because other subclasses could be added later to the family of types, and objects of those new subclasses would also work with the existing code.

--------------------------------------------

**What does it mean that a method or field is “static”?** -

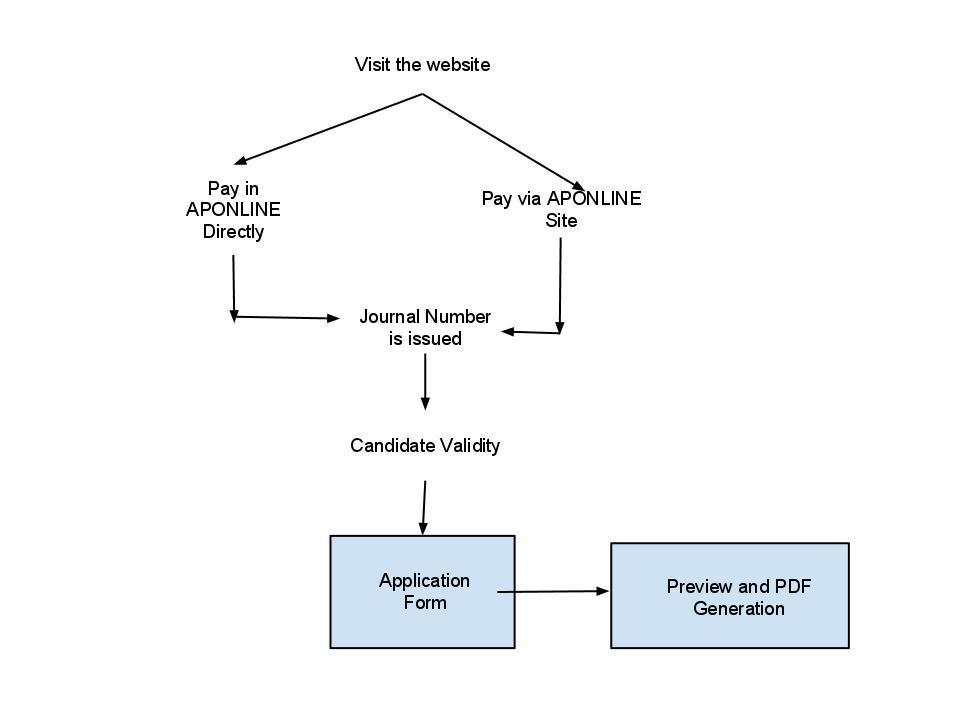
Static variables and methods are instantiated only once per class. In other words they are class variables, not instance variables. If you change the value of a static variable in a particular object, the value of that variable changes for all instances of that class. Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too). That’s how library methods like System.out.println() work. out is a static field in the java.lang.System class.

-----------------------------------------------------------------------------

**How do I convert a numeric IP address like 192.18.97.39 into a hostname like java.sun.com?**

String hostname = InetAddress.getByName("192.18.97.39").getHostName();

**Architecture of CGG Project:**



-------------------------------------------------------

**Difference between DBMS and RDBMS:**

DBMS:

DBMS is defined as the software program that is used to manage all the databases that are stored on the network or system hard disk. There are different types of database management systems and some of them are configured for specific purposes.DBMS is available in different forms as a tool that is used to manage databases. Some popular DBMS solutions include DB2, Oracle, FileMaker and Microsoft Access. Using these products, privileges or rights can be created that can be specific to particular users. It means that the administrators of the database can grant specific rights to some users or assign different levels of administration.Every DBMS has some fundamental elements. First is the implementation of the modeling language which defines the language used for each database. Second, DBMS also administers the data structures. Data query language is the third element of a DBMS. Data structures work with data query language in order to make sure that irrelevant data cannot be entered into the database used in the system.

RDBMS:

The database system in which the relationships among different tables are maintained is called Relational Database Management System. Both RDBMS and DBMS are used to store information in physical database.RDBMS solution is required when large amounts of data are to be stored as well as maintained. A relational data model consists of indexes, keys, foreign keys, tables and their relationships with other tables. In 1970s, Edgar Frank Codd introduced the theory of relational database. Twelve rules were defined by Codd for this relational theory or model. Relationships among different types of data Is the main requirement of the relational model. RDMS can be termed as the next generation of database management system. DBMS is used as a base model in order to store data in a relational database system. However, complex business [applications](http://www.differencebetween.com/category/technology/it/applications/) use RDBMS rather than DBMS.

Any DBMS to be called as RDBMS it needs to support some of the CODD rules:

CODD Rules:

1. Information rule.

2. Data Distribution rule.

3. Non Sub version rule.

4. Rule of guaranteed access.

5. Logical Independence rule.

6. Data descriptive rule.

7. Data integrity rule.

8. Data Sub language rule.

9. High level of insert,update and delete.

10. Physical Independence rule.

11. Systematic treatment of null values.

12. View Updation rule.

For explanation of the above rules use the below link:

<http://en.wikipedia.org/wiki/Codd%27s_12_rules>

------------------------------------------------------------------------------------------------------------------------------------------------------------

**Implicit objects in a JSP.**

application --- javax.servlet.ServletContext

config --- javax.servlet.ServletConfig

exception --- java.lang.Throwable

out --- javax.servlet.jsp.JspWriter

page --- java.lang.Object

PageContext --- javax.servlet.jsp.PageContext

Request --- javax.servlet.ServletRequest

Response --- javax.servlet.ServletResponse

session --- javax.servlet.http.HttpSession

------------------------------------------------------------------------------------------------------------------------------------------------------------

**Difference between Forward and sendRedirect**

Forward

Since forward() method of RequestDispatcher is handled on the server , therefore the request and its associated session are available to the forwarded resource and you can pass data between them using request.setAttribute(). forward() separates the responsibilities for handling the requests among several components. This method generally sends a request and response object to resources (servlets or JSP's) of the same ServletContext. You can also decide to forward to one page in one condition and a different page in another. RequestDispatcher transfers control immediately whenever the forward() method is called to the resource referenced by the RequestDispatcher.

A forward is performed internally by the container the browser is completely unaware that it has taken place, so its original URL remains intact and any browser reload of the resulting page will simply repeat the original request, with the original URL. When the forward is done, the original request and response objects are transfered along with additional parameters if needed.

sendRedirect

The transfer of control is delegated to the browser by the container. That is, the redirect sends a header back to the browser / client. This header contains the resource url to be redirected by the browser. Then the browser initiates a new request to the given url. Since it is a new request, the old request and response object is lost. It is like keying in a new URL in the browser. A sendRedirect() also updates the browser history and transfers control only when the whole service method completes. There is only one way to pass data is through the session or using web parameters.

RequestDispatcher.forward() and PageContext.forward() are effectively the same. PageContext.forward is a helper method that calls the RequestDispatcher method.

When you invoke RequestDispatcher.include(), the servlet engine transfers control of this HTTP request internally from your current servlet or JSP to another servlet or JSP or static file, while invoking response.sendRedirect() method sends an HTTP response to the browser to make another request at a different URL.

A redirect is a two step process, where the web application instructs the browser to fetch a second URL, which differs from the original. A browser reload of the second URL will not repeat the original request, but will rather fetch the second URL.Redirect is slower than a forward, since it requires two browser requests, not one . Objects placed in the original request scope are not available to the second request

For example, sendRedirect can transfer control from http://javapapers.com to http://anydomain.com but forward cannot do this.

‘session’ is not lost in both forward and redirect.

To identify difference between forward and sendRedirect keep an eye on the address bar of your browser, in forward, you will not see the forwarded address in redirect, you can see the redirected address.

------------------------------------------------------------------------------------------------------------------------------------------------------------

**What happens if we get an exception in finally block.**

The code in the finally block is intended to be "cleanup" code, so we can handle this situation in the following way:

public int myExeptionMethod() throws IOException, NumberFormatException {

...

. . .

. . .

try {

-----------

---------

}

finally {

**try {**

**}**

catch (IOException e) {

}

}

}

What will be the result of the following code:

public class Tester {

public static void main(String[] args) {

System.out.print("1");

try {

System.out.print("2");

System.exit(0);

} finally {

System.out.print("3");

}

}

}

Ans : 12 : System.exit(0) will cause the program to exit and finally block will not be executed

**Write a programme to print the reverse of a number.**

**Reason For Change:**

1. Onsite Oppurtunity very unlikely in the present company.

2.

**When do we use List and when do we use ‘Vector’ ?**

**struts-config.xml example tags:  
  
Reason for usage of generics:**

**New Features in Java1.5 and later versions:**

**ArrayList vs Lists:**

**Arrays vs ArrayLists:**

**Eclipse vs Myeclipse:**

Eclipse and Myeclipse are somewhat similar in a first take. However, once you get to use it more often, you will realize that they are absolutely different. The Eclipse plug-in entails an architectural pattern for creating an application from constituent parts. It is an extensible platform used for creating IDEs. In addition, it has given a core of services for controlling a pack of tools working together to support programming tasks. There are tool builders that contribute to the Eclipse platform. They are shared by wrapping their tools in pluggable components. This will conform to the platform. Also, the basic mechanism of the extensibility is brought about by the new plug-ins in the new processing elements to plug-ins that already exists. Despite the fact that it focuses on building IDEs, its concepts and implications support a general model for coming up with an application from different parts created by different companies.

With regards to the Myeclipse enterprise workbench, it is a full-featured platform and an enterprise-class plug-in. It has a tool fit for developing [software](http://www.differencebetween.net/category/technology/software-technology/) applications and systems that support the full life cycle of application development. Myeclipse has so much to offer. It has passed the open-industry standards, and it even redefined the pricing of the software, support, and delivery release cycles. It has created a complete application development environment for J2EE WEB, XML, UML and databases. Not only that, it has the most comprehensive selection of application [server](http://www.differencebetween.net/technology/software-technology/difference-between-application-server-and-web-server/) connectors with over 25 target environments. It has the capacity to optimize development, deployment testing, and even portability.

The Eclipse plug-in model is mainly a part the gives a certain type of service inside the context of the Eclipse workbench. The meaning of “component” here [would](http://www.differencebetween.net/language/difference-between-would-and-should/) [mean](http://www.differencebetween.net/science/difference-between-sample-mean-and-population-mean/) the object that has been configured into a [system](http://www.differencebetween.net/miscellaneous/difference-between-caste-system-and-class-system/) at a certain deployment time. The runtime of Eclipse will give an infrastructure the necessary support to activate and operate a set of plug-ins working hand in hand. This is necessary so that there’s a seamless environment for the development of activities. As Eclipse runs for an instance, a plug-in will have to be connected in an instance of some plug-in runtime class or more commonly known as the plug-in class. The plug-in class will create a necessary configuration and support for the management. The plug-in class will then have to extend to org.eclipse.core.runtime. This will be the abstract class that can generate facilities for the [management](http://www.differencebetween.net/editor-pick/difference-between-management-and-control/) of plug-ins. The plug-in installation would involve the plug-ins folder wherein you will see individual plug-ins in it. Such plug-ins will be inscribed in an [XML](http://www.differencebetween.net/technology/difference-between-xml-and-xsd/) type of file. This will allow the file to inform Eclipse runtime what plug-ins need to be activated.

The Myeclipse model works by offering users the versatility to choose the technology required at every application tier. It will also offer optional technology bundles and the access to velocity templates for the generation of [codes](http://www.differencebetween.net/science/health/difference-between-cpt-and-icd-codes/) and the addition of a third-party commercial. It also has the OSS tools for further development. What is more is that there is an improvement in the developer activity. It enhances your Eclipse experience by providing you a [Java](http://www.differencebetween.net/technology/difference-between-python-and-java/) EE/J2EE database and the rich-client development environment. The workbench also has an embedded Tomcat server giving you the option to have a sandbox. RAD, UML, POJOs, Web 2.0 is also provided for you. MyEclipse has maximized its potential to effectively provide you the tools you need for every size development project. It can be from a company of one to the largest multi-national firms. It is a credible software development tool used by so many elite companies like IBM, United Airlines, Glaxo Smith Kline, and even in the European Parliament.

Summary:

1. Eclipse is a plug-in for an architectural pattern. MyEclipse is a full-featured platform for software development application.

2. Eclipse is part of the context of an Eclipse workbench. The plug-in configures into a system at a certain runtime.

3. MyEclipse works by offering optional technology bundles.

**What is interface and its use?**

- Interface is similar to a class which may contain method’s signature only but not bodies and it is a formal set of methods and constants declarations that must be defined by the class that implements it. Interfaces are useful for:

a)Declaring methods that one or more classes are expected to implement.

b)Capturing similarities between unrelated classes without forcing a class relationship.

c)Determining an object’s programming interface without revealing the actual body of the class.

**What are the similarities/difference between an Abstract class and Interface**?  
Answer:  Differences are as follows:

Interfaces provide a form of multiple inheritance. A class can extend only one other class.

Interfaces are limited to public methods and constants with no implementation. Abstract classes can have a partial implementation, protected parts, static methods, etc.

A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.

Interfaces are slow as it requires extra indirection to to find corresponding method in in the actual class. Abstract classes are fast.

Neither Abstract classes or Interface can be instantiated.

Abstract classes are very much useful when there is a some functionality across various classes. Interfaces are well suited for the classes which varies in functionality but with the same method signatures.

**Can a abstract method have the static qualifier?** - No

**Explain the Polymorphism principle**

Answer: The meaning of Polymorphism is something like one name many forms. Polymorphism enables one entity to be used as as

general category for different types of actions. The specific action is determined by the exact nature of the situation. The

concept of polymorphism can be explained as "one interface, multiple methods".

**Different forms of Polymorphism:**  
Polymorphism exists in three distinct forms in Java:

Method overloading

Method overriding through inheritance

Method overriding through the Java interface.

**What interface do you implement to do the sorting? --**  Comparable

**What is the eligibility for a object to get cloned?**  It must implement the Cloneable interface

**What is hash-collision in Hashtable and how it is handled in Java?** - Two different keys with the same hash value. Two different entries will be kept in a single hash bucket to avoid the collision.

**What is the super class of Hashtable? – Dictionary**

**What is the use of serializable?** - To persist the state of an object into any permanent storage device.

**What is transient variable?**- Transient variable can't be serialized. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStream the value of the variable becomes null.

**What are the different level lockings using the synchronization keyword?** - Class level lock , Object level lock, Method level lock, Block level lock

**What are the different types of qualifier and what is the default qualifier?** - public, protected, private, package (default)

**What do you mean by a Classloader?** - Classloader is the one which loads the classes into the JVM.

**What are the implicit packages that need not get imported into a class file?** - java.lang

**What is a thread?** - Thread is a block of code which can execute concurrently with other threads in the JVM.

**What is the difference between procedural and object-oriented programs?**- a) In procedural program, programming logic follows certain procedures and the instructions are executed one after another. In OOP program, unit of program is object, which is nothing but combination of data and code. b) In procedural program, data is exposed to the whole program whereas in OOPs program, it is accessible with in the object and which in turn assures the security of the code.

**What are Encapsulation, Inheritance and Polymorphism?**- Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse. Inheritance is the process by which one object acquires the properties of another object. Polymorphism is the feature that allows one interface to be used for general class actions.

**What is the difference between Assignment and Initialization?**- Assignment can be done as many times as desired whereas initialization can be done only once.

**What is OOPs?**  Object oriented programming organizes a program around its data, i. e. , objects and a set of well defined interfaces to that data. An object-oriented program can be characterized as data controlling access to code.

**What are Class, Constructor and Primitive data types?** Class is a template for multiple objects with similar features and it is a blue print for objects. It defines a type of object according to the data the object can hold and the operations the object can perform. Constructor is a special kind of method that determines how an object is initialized when created. Primitive data types are 8 types and they are: byte, short, int, long, float, double, boolean, char.

**What is an Object and how do you allocate memory to it?** - Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

**What is the difference between constructor and method?**- Constructor will be automatically invoked when an object is created whereas method has to be called explicitly.

**What are methods and how are they defined?**- Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method. A method’s signature is a combination of the first three parts mentioned above.

**What is the use of bin and lib in JDK?**- Bin contains all tools such as javac, appletviewer, awt tool, etc., whereas lib contains API and all packages.

**What is casting?**- Casting is used to convert the value of one type to another.

**How many ways can an argument be passed to a subroutine and explain them?**- An argument can be passed in two ways. They are passing by value and passing by reference. Passing by value: This method copies the value of an argument into the formal parameter of the subroutine. Passing by reference: In this method, a reference to an argument (not the value of the argument) is passed to the parameter.

**What is the difference between an argument and a parameter?**- While defining method, variables passed in the method are called parameters. While using those methods, values passed to those variables are called arguments.

**What are different types of access modifiers?**- public: Any thing declared as public can be accessed from anywhere. private: Any thing declared as private can’t be seen outside of its class. protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages. default modifier : Can be accessed only to classes in the same package.

**What is final, finalize() and finally?**- final : final keyword can be used for class, method and variables. A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods. A final method can’t be overridden. A final variable can’t change from its initialized value. finalize() : finalize() method is used just before an object is destroyed and can be called just prior to garbage collection. finally : finally, a key word used in exception handling, creates a block of code that will be executed after a try/catch block has completed and before the code following the try/catch block. The finally block will execute whether or not an exception is thrown. For example, if a method opens a file upon exit, then you will not want the code that closes the file to be bypassed by the exception-handling mechanism. This finally keyword is designed to address this contingency.

**What is Garbage Collection and how to call it explicitly?**- When an object is no longer referred to by any variable, java automatically reclaims memory used by that object. This is known as garbage collection. System. gc() method may be used to call it explicitly.

**What is finalize() method?**- finalize () method is used just before an object is destroyed and can be called just prior to garbage collection.

**What are Transient and Volatile Modifiers?**- Transient: The transient modifier applies to variables only and it is not stored as part of its object’s Persistent state. Transient variables are not serialized. Volatile: Volatile modifier applies to variables only and it tells the compiler that the variable modified by volatile can be changed unexpectedly by other parts of the program.

**What is method overloading and method overriding?**- Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding : When a method in a class having the same method name with same arguments is said to be method overriding.

**What is difference between overloading and overriding?**- a) In overloading, there is a relationship between methods available in the same class whereas in overriding, there is relationship between a superclass method and subclass method. b) Overloading does not block inheritance from the superclass whereas overriding blocks inheritance from the superclass. c) In overloading, separate methods share the same name whereas in overriding, subclass method replaces the superclass. d) Overloading must have different method signatures whereas overriding must have same signature.

**What is meant by Inheritance and what are its advantages?**- Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

**What is the difference between this() and super()?**- this() can be used to invoke a constructor of the same class whereas super() can be used to invoke a super class constructor.

**What is the difference between superclass and subclass?**- A super class is a class that is inherited whereas sub class is a class that does the inheriting.

**What modifiers may be used with top-level class?**- public, abstract and final can be used for top-level class.

**What are inner class and anonymous class?**- Inner class : classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private. Anonymous class : Anonymous class is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.

**What is a package?**- A package is a collection of classes and interfaces that provides a high-level layer of access protection and name space management.

**What is the difference between Integer and int?**- a) Integer is a class defined in the java. lang package, whereas int is a primitive data type defined in the Java language itself. Java does not automatically convert from one to the other. b) Integer can be used as an argument for a method that requires an object, whereas int can be used for calculations.

**What is a Marker Interface?** An interface with no methods. Example: Serializable, Remote, Cloneable

**What is a cloneable interface and how many methods does it contain?**- It is not having any method because it is a TAGGED or MARKER interface.

**What is the difference between abstract class and interface?**- a) All the methods declared inside an interface are abstract whereas abstract class must have at least one abstract method and others may be concrete or abstract. b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods. c) Abstract class must have subclasses whereas interface can’t have subclasses.

**Can you have an inner class inside a method and what variables can you access?**- Yes, we can have an inner class inside a method and final variables can be accessed.

**What is the difference between String and String Buffer?**- a) String objects are constants and immutable whereas StringBuffer objects are not. b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

**What is the difference between Array and vector?**- Array is a set of related data type and static whereas vector is a growable array of objects and dynamic.

**What is the difference between exception and error?**- The exception class defines mild error conditions that your program encounters. Exceptions can occur when trying to open the file, which does not exist, the network connection is disrupted, operands being manipulated are out of prescribed ranges, the class file you are interested in loading is missing. The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program terminate when such an error is encountered.

**What is the difference between lightweight and heavyweight component?** - Lightweight components reuses its parents graphical units. Heavyweight components goes with the native graphical unit for every component. Lightweight components are faster than the heavyweight components.

**What are the ways in which you can instantiate a thread?** - Using Thread class By implementing the Runnable interface and giving that handle to the Thread class.

**What are the states of a thread?** - 1. New 2. Runnable 3. Not Runnable 4. Dead

**What is a socket?** - A socket is an endpoint for communication between two machines.

**What are the threads will start, when you start the java program?** - Finalizer, Main, Reference Handler, Signal Dispatcher

**What is the difference between process and thread?**- Process is a program in execution whereas thread is a separate path of execution in a program.

**What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?**- Multithreading is the mechanism in which more than one thread run independent of each other within the process. wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class. wait() : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state. notify() or notifyAll() : To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.

**What is the class and interface in java to create thread and which is the most advantageous method?**- Thread class and Runnable interface can be used to create threads and using Runnable interface is the most advantageous method to create threads because we need not extend thread class here.

**What are the states associated in the thread?**- Thread contains ready, running, waiting and dead states.

**What is synchronization?**- Synchronization is the mechanism that ensures that only one thread is accessed the resources at a time.

**When you will synchronize a piece of your code?**- When you expect your code will be accessed by different threads and these threads may change a particular data causing data corruption.

**What is deadlock?**- When two threads are waiting each other and can’t precede the program is said to be deadlock.

**What is daemon thread and which method is used to create the daemon thread?**- Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

**Are there any global variables in Java, which can be accessed by other part of your program?**- No, it is not the main method in which you define variables. Global variables is not possible because concept of encapsulation is eliminated here.

**What are wrapper classes?**- Wrapper classes are classes that allow primitive types to be accessed as objects

**What is Collection API?**  
Answer: The Collection API is a set of classes and interfaces that support operation on collections of objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.   
Example of classes: HashSet, HashMap, ArrayList, LinkedList, TreeSet and TreeMap.  
Example of interfaces: Collection, Set, List and Map.

**What are Vector, Hashtable, LinkedList and Enumeration?**- Vector : The Vector class provides the capability to implement a growable array of objects. Hashtable : The Hashtable class implements a Hashtable data structure. A Hashtable indexes and stores objects in a dictionary using hash codes as the object’s keys. Hash codes are integer values that identify objects. LinkedList: Removing or inserting elements in the middle of an array can be done using LinkedList. A LinkedList stores each object in a separate link whereas an array stores object references in consecutive locations. Enumeration: An object that implements the Enumeration interface generates a series of elements, one at a time. It has two methods, namely hasMoreElements() and nextElement(). HasMoreElemnts() tests if this enumeration has more elements and nextElement method returns successive elements of the series.

**What is the difference between set and list?**- Set stores elements in an unordered way but does not contain duplicate elements, whereas list stores elements in an ordered way but may contain duplicate elements.

**Is Iterator a Class or Interface? What is its use?** Iterator is an interface which is used to step through the elements of a Collection

**What is a stream and what are the types of Streams and classes of the Streams?**- A Stream is an abstraction that either produces or consumes information. There are two types of Streams and they are: Byte Streams: Provide a convenient means for handling input and output of bytes. Character Streams: Provide a convenient means for handling input & output of characters. Byte Streams classes: Are defined by using two abstract classes, namely InputStream and OutputStream. Character Streams classes: Are defined by using two abstract classes, namely Reader and Writer.

**What is the difference between Reader/Writer and InputStream/Output Stream?**- The Reader/Writer class is character-oriented and the InputStream/OutputStream class is byte-oriented.

**What is an I/O filter?**- An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

**What is serialization and deserialization?**- Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

**What is JDBC?**- JDBC is a set of Java API for executing SQL statements. This API consists of a set of classes and interfaces to enable programs to write pure Java Database applications.

**What are drivers available?**- a) JDBC-ODBC Bridge driver b) Native API Partly-Java driver c) JDBC-Net Pure Java driver d) Native-Protocol Pure Java driver

**What is the difference between JDBC and ODBC?**- a) OBDC is for Microsoft and JDBC is for Java applications. b) ODBC can’t be directly used with Java because it uses a C interface. c) ODBC makes use of pointers which have been removed totally from Java. d) ODBC mixes simple and advanced features together and has complex options for simple queries. But JDBC is designed to keep things simple while allowing advanced capabilities when required. e) ODBC requires manual installation of the ODBC driver manager and driver on all client machines. JDBC drivers are written in Java and JDBC code is automatically installable, secure, and portable on all platforms. f) JDBC API is a natural Java interface and is built on ODBC. JDBC retains some of the basic features of ODBC.

**Does the JDBC-ODBC Bridge support multiple concurrent open statements per connection?** - No

**What are the types of JDBC Driver Models and explain them?**- There are two types of JDBC Driver Models and they are: a) Two tier model and b) Three tier model Two tier model: In this model, Java applications interact directly with the database. A JDBC driver is required to communicate with the particular database management system that is being accessed. SQL statements are sent to the database and the results are given to user. This model is referred to as client/server configuration where user is the client and the machine that has the database is called as the server. Three tier model: A middle tier is introduced in this model. The functions of this model are: a) Collection of SQL statements from the client and handing it over to the database, b) Receiving results from database to the client and c) Maintaining control over accessing and updating of the above.

**What are the steps involved for making a connection with a database or how do you connect to a database?**a) Loading the driver : To load the driver, Class. forName() method is used. Class. forName(”sun. jdbc. odbc. JdbcOdbcDriver”); When the driver is loaded, it registers itself with the java. sql. DriverManager class as an available database driver. b) Making a connection with database: To open a connection to a given database, DriverManager. getConnection() method is used. Connection con = DriverManager. getConnection (”jdbc:odbc:somedb”, “user”, “password”); c) Executing SQL statements : To execute a SQL query, java. sql. statements class is used. createStatement() method of Connection to obtain a new Statement object. Statement stmt = con. createStatement(); A query that returns data can be executed using the executeQuery() method of Statement. This method executes the statement and returns a java. sql. ResultSet that encapsulates the retrieved data: ResultSet rs = stmt. executeQuery(”SELECT \* FROM some table”); d) Process the results : ResultSet returns one row at a time. Next() method of ResultSet object can be called to move to the next row. The getString() and getObject() methods are used for retrieving column values: while(rs. next()) { String event = rs. getString(”event”); Object count = (Integer) rs. getObject(”count”);

**What type of driver did you use in project?** Type 4 driver

**What are the types of statements in JDBC?**- Statement: to be used createStatement() method for executing single SQL statement PreparedStatement — To be used preparedStatement() method for executing same SQL statement over and over. CallableStatement — To be used prepareCall() method for multiple SQL statements over and over.

**What is stored procedure?**- Stored procedure is a group of SQL statements that forms a logical unit and performs a particular task. Stored Procedures are used to encapsulate a set of operations or queries to execute on database. Stored procedures can be compiled and executed with different parameters and results and may have any combination of input/output parameters.

**How to create and call stored procedures?**- To create stored procedures: Create procedure procedurename (specify in, out and in out parameters) BEGIN Any multiple SQL statement; END; To call stored procedures: CallableStatement csmt = con. prepareCall(”{call procedure name(?,?)}”); csmt. registerOutParameter(column no. , data type); csmt. setInt(column no. , column name) csmt. execute();

**What is servlet?**- Servlets are modules that extend request/response-oriented servers, such as java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company’s order database.

**What are the classes and interfaces for servlets?**- There are two packages in servlets and they are javax. servlet and

**What is the difference between doPost and doGet methods?**- a) doGet() method is used to get information, while doPost() method is used for posting information. b) doGet() requests can’t send large amount of information and is limited to 240-255 characters. However, doPost()requests passes all of its data, of unlimited length. c) A doGet() request is appended to the request URL in a query string and this allows the exchange is visible to the client, whereas a doPost() request passes directly over the socket connection as part of its HTTP request body and the exchange are invisible to the client.

**What is the life cycle of a servlet?**- Each Servlet has the same life cycle: a) A server loads and initializes the servlet by init () method. b) The servlet handles zero or more client’s requests through service() method. c) The server removes the servlet through destroy() method.

**Who is loading the init() method of servlet?**- Web server

**What are the different servers available for developing and deploying Servlets?**- a) Java Web Server b) JRun g) Apache Server h) Netscape Information Server i) Web Logic

**How many ways can we track client and what are they?**- The servlet API provides two ways to track client state and they are: a) Using Session tracking and b) Using Cookies.

**What is session tracking and how do you track a user session in servlets?**- Session tracking is a mechanism that servlets use to maintain state about a series requests from the same user across some period of time. The methods used for session tracking are: a) User Authentication - occurs when a web server restricts access to some of its resources to only those clients that log in using a recognized username and password. b) Hidden form fields - fields are added to an HTML form that are not displayed in the client’s browser. When the form containing the fields is submitted, the fields are sent back to the server. c) URL rewriting - every URL that the user clicks on is dynamically modified or rewritten to include extra information. The extra information can be in the form of extra path information, added parameters or some custom, server-specific URL change. d) Cookies - a bit of information that is sent by a web server to a browser and which can later be read back from that browser. e) HttpSession- places a limit on the number of sessions that can exist in memory. This limit is set in the session. maxresidents property.

**What is Server-Side Includes (SSI)?**- Server-Side Includes allows embedding servlets within HTML pages using a special servlet tag. In many servlets that support servlets, a page can be processed by the server to include output from servlets at certain points inside the HTML page. This is accomplished using a special internal SSINCLUDE, which processes the servlet tags. SSINCLUDE servlet will be invoked whenever a file with an. shtml extension is requested. So HTML files that include server-side includes must be stored with an . shtml extension.

**What are cookies and how will you use them?**- Cookies are a mechanism that a servlet uses to have a client hold a small amount of state-information associated with the user. a) Create a cookie with the Cookie constructor: public Cookie(String name, String value) b) A servlet can send a cookie to the client by passing a Cookie object to the addCookie() method of HttpServletResponse: public void HttpServletResponse. addCookie(Cookie cookie) c) A servlet retrieves cookies by calling the getCookies() method of HttpServletRequest: public Cookie[ ] HttpServletRequest. getCookie().

**Is it possible to communicate from an applet to servlet and how many ways and how?**- Yes, there are three ways to communicate from an applet to servlet and they are: a) HTTP Communication(Text-based and object-based) b) Socket Communication c) RMI Communication

**What is connection pooling?**- With servlets, opening a database connection is a major bottleneck because we are creating and tearing down a new connection for every page request and the time taken to create connection will be more. Creating a connection pool is an ideal approach for a complicated servlet. With a connection pool, we can duplicate only the resources we need to duplicate rather than the entire servlet. A connection pool can also intelligently manage the size of the pool and make sure each connection remains valid. A number of connection pool packages are currently available. Some like DbConnectionBroker are freely available from Java Exchange Works by creating an object that dispenses connections and connection Ids on request. The ConnectionPool class maintains a Hastable, using Connection objects as keys and Boolean values as stored values. The Boolean value indicates whether a connection is in use or not. A program calls getConnection() method of the ConnectionPool for getting Connection object it can use; it calls returnConnection() to give the connection back to the pool.

**Why should we go for interservlet communication?**- Servlets running together in the same server communicate with each other in several ways. The three major reasons to use interservlet communication are: a) Direct servlet manipulation - allows to gain access to the other currently loaded servlets and perform certain tasks (through the ServletContext object) b) Servlet reuse - allows the servlet to reuse the public methods of another servlet. c) Servlet collaboration - requires to communicate with each other by sharing specific information (through method invocation)

**Is it possible to call servlet with parameters in the URL?**- Yes. You can call a servlet with parameters in the syntax as (?Param1 = xxx || m2 = yyy).

**What is Servlet chaining?**- Servlet chaining is a technique in which two or more servlets can cooperate in servicing a single request. In servlet chaining, one servlet’s output is piped to the next servlet’s input. This process continues until the last servlet is reached. Its output is then sent back to the client.

**How do servlets handle multiple simultaneous requests?**- The server has multiple threads that are available to handle requests. When a request comes in, it is assigned to a thread, which calls a service method (for example: doGet(), doPost() and service()) of the servlet. For this reason, a single servlet object can have its service methods called by many threads at once.

**What is Inet address?**- Every computer connected to a network has an IP address. An IP address is a number that uniquely identifies each computer on the Net. An IP address is a 32-bit number.

**What is Domain Naming Service(DNS)?**- It is very difficult to remember a set of numbers(IP address) to connect to the Internet. The Domain Naming Service(DNS) is used to overcome this problem. It maps one particular IP address to a string of characters. For example, www. mascom. com implies com is the domain name reserved for US commercial sites, moscom is the name of the company and www is the name of the specific computer, which is mascom’s server.

**What is RMI and steps involved in developing an RMI object?**- Remote Method Invocation (RMI) allows java object that executes on one machine and to invoke the method of a Java object to execute on another machine. The steps involved in developing an RMI object are: a) Define the interfaces b) Implementing these interfaces c) Compile the interfaces and their implementations with the java compiler d) Compile the server implementation with RMI compiler e) Run the RMI registry f) Run the application

**What is RMI architecture?**- RMI architecture consists of four layers and each layer performs specific functions: a) Application layer - contains the actual object definition. b) Proxy layer - consists of stub and skeleton. c) Remote Reference layer - gets the stream of bytes from the transport layer and sends it to the proxy layer. d) Transportation layer - responsible for handling the actual machine-to-machine communication.

**What is the protocol used by RMI?** - RMI-IIOP

**what is UnicastRemoteObject?**- All remote objects must extend UnicastRemoteObject, which provides functionality that is needed to make objects available from remote machines.

**Explain the methods, rebind() and lookup() in Naming class?**- rebind() of the Naming class(found in java. rmi) is used to update the RMI registry on the server machine. Naming. rebind(”AddSever”, AddServerImpl); lookup() of the Naming class accepts one argument, the rmi URL and returns a reference to an object of type AddServerImpl.

**What is a Java Bean?**- A Java Bean is a software component that has been designed to be reusable in a variety of different environments.

**What is a Jar file?**- Jar file allows to efficiently deploying a set of classes and their associated resources. The elements in a jar file are compressed, which makes downloading a Jar file much faster than separately downloading several uncompressed files. The package java. util. zip contains classes that read and write jar files.

**What is BDK?**- BDK, Bean Development Kit is a tool that enables to create, configure and connect a set of set of Beans and it can be used to test Beans without writing a code.

**What is JSP?**- JSP is a dynamic scripting capability for web pages that allows Java as well as a few special tags to be embedded into a web file (HTML/XML, etc). The suffix traditionally ends with .jsp to indicate to the web server that the file is a JSP files. JSP is a server side technology - you can’t do any client side validation with it. The advantages are: a) The JSP assists in making the HTML more functional. Servlets on the other hand allow outputting of HTML but it is a tedious process. b) It is easy to make a change and then let the JSP capability of the web server you are using deal with compiling it into a servlet and running it.

**What are JSP scripting elements?**- JSP scripting elements lets to insert Java code into the servlet that will be generated from the current JSP page. There are three forms: a) Expressions of the form <%= expression %> that are evaluated and inserted into the output, b) Scriptlets of the formthat are inserted into the servlet’s service method, and c) Declarations of the form <%! Code %>that are inserted into the body of the servlet class, outside of any existing methods.

**What are JSP Directives?**- A JSP directive affects the overall structure of the servlet class. It usually has the following form:<%@ directive attribute=”value” %> However, you can also combine multiple attribute settings for a single directive, as follows:<%@ directive attribute1=”value1″ attribute 2=”value2″ . . . attributeN =”valueN” %> There are two main types of directive: page, which lets to do things like import classes, customize the servlet superclass, and the like; and include, which lets to insert a file into the servlet class at the time the JSP file is translated into a servlet

**What are Predefined variables or implicit objects?**- To simplify code in JSP expressions and scriptlets, we can use eight automatically defined variables, sometimes called implicit objects. They are request, response, out, session, application, config, pageContext, and page.

**What are JSP ACTIONS?**- JSP actions use constructs in XML syntax to control the behavior of the servlet engine. You can dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin. Available actions include: jsp:include - Include a file at the time the page is requested. jsp:useBean - Find or instantiate a JavaBean. jsp:setProperty - Set the property of a JavaBean. jsp:getProperty - Insert the property of a JavaBean into the output. jsp:forward - Forward the requester to a newpage. Jsp: plugin - Generate browser-specific code that makes an OBJECT or EMBED

**How do you pass data (including JavaBeans) to a JSP from a servlet?**- (1) Request Lifetime: Using this technique to pass beans, a request dispatcher (using either “include” or forward”) can be called. This bean will disappear after processing this request has been completed. Servlet: request. setAttribute(”theBean”, myBean); RequestDispatcher rd = getServletContext(). getRequestDispatcher(”thepage. jsp”); rd. forward(request, response); JSP PAGE:<jsp: useBean id=”theBean” scope=”request” class=”. . . . . ” />(2) Session Lifetime: Using this technique to pass beans that are relevant to a particular session (such as in individual user login) over a number of requests. This bean will disappear when the session is invalidated or it times out, or when you remove it. Servlet: HttpSession session = request. getSession(true); session. putValue(”theBean”, myBean); /\* You can do a request dispatcher here, or just let the bean be visible on the next request \*/ JSP Page:<jsp:useBean id=”theBean” scope=”session” class=”. . . ” /> 3) Application Lifetime: Using this technique to pass beans that are relevant to all servlets and JSP pages in a particular app, for all users. For example, I use this to make a JDBC connection pool object available to the various servlets and JSP pages in my apps. This bean will disappear when the servlet engine is shut down, or when you remove it. Servlet: GetServletContext(). setAttribute(”theBean”, myBean); JSP PAGE:<jsp:useBean id=”theBean” scope=”application” class=”. . . ” />

**How can I set a cookie in JSP?**- response. setHeader(”Set-Cookie”, “cookie string”); To give the response-object to a bean, write a method setResponse (HttpServletResponse response) - to the bean, and in jsp-file:<% bean. setResponse (response); %>

**How can I delete a cookie with JSP?**- Say that I have a cookie called “foo, ” that I set a while ago & I want it to go away. I simply: <% Cookie killCookie = new Cookie(”foo”, null); KillCookie. setPath(”/”); killCookie. setMaxAge(0); response. addCookie(killCookie); %>

**What is Jakarta Struts Framework?**  
A: Jakarta Struts is [open source](http://www.roseindia.net/interviewquestions/jakartastrutsinterviewquestions.shtml##) implementation of MVC (Model-View-Controller) pattern for the development of web based applications. Jakarta Struts is robust architecture and can be used for the development of application of any size. Struts framework makes it much easier to design scalable, reliable Web applications with [Java](http://www.roseindia.net/interviewquestions/jakartastrutsinterviewquestions.shtml##).

How you will make available any Message Resources Definitions file to the Struts Framework Environment?A: Message Resources Definitions file are simple .properties files and these files contains the messages that can be used in the struts project. Message Resources Definitions files can be added to the struts-config.xml file through <message-resources /> tag.  
Example:  
<message-resources parameter="MessageResources" />

Q: What is Action Class?  
A: The Action is part of the controller. The purpose of Action Class is to translate the HttpServletRequest to the business logic. To use the Action, we need to  Subclass and overwrite the execute()  method. The ActionServlet (commad) passes the parameterized class to Action Form using the execute() method. There should be no database interactions in the action. The action should receive the request, call business objects (which then handle database, or interface with J2EE, etc) and then determine where to go next. Even better, the business objects could be handed to the action at runtime (IoC style) thus removing any dependencies on the model.   The return type of the execute method is ActionForward which is used by the Struts Framework to forward the request to the file as per the value of the returned ActionForward object.

Q: Write code of any Action Class?  
A: Here is the code of Action Class that returns the ActionForward object.  
TestAction.java

| | package roseindia.net;  import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;  import org.apache.struts.action.Action; import org.apache.struts.action.ActionForm; import org.apache.struts.action.ActionForward; import org.apache.struts.action.ActionMapping;  public class TestAction extends Action {   public ActionForward execute(     ActionMapping mapping,     ActionForm form,     HttpServletRequest request,     HttpServletResponse response) throws Exception{       return mapping.findForward("testAction");   } } | | --- | |
| --- | --- |

Q: What is ActionForm?  
A: An ActionForm is a JavaBean that extends org.apache.struts.action.ActionForm. ActionForm maintains the session state for [web application](http://www.roseindia.net/interviewquestions/jakartastrutsinterviewquestions.shtml##) and the ActionForm object is automatically populated on the server side with data entered from a form on the client side.

Q: What is Struts Validator Framework?  
A: Struts Framework provides the functionality to validate the form data. It can be use to validate the data on the users browser as well as on the server side. Struts Framework emits the [java scripts](http://www.roseindia.net/interviewquestions/jakartastrutsinterviewquestions.shtml##) and it can be used validate the form data on the client browser. Server side validation of form can be accomplished by sub classing your From Bean withDynaValidatorForm class.

The Validator framework was developed by David Winterfeldt as third-party add-on to Struts. Now the Validator framework is a part of Jakarta Commons project and it can be used with or without Struts. The Validator framework comes integrated with the Struts Framework and can be used without doing any extra settings.

Q. Give the Details of XML files used in Validator Framework?  
A: The Validator Framework uses two XML [configuration files](http://www.roseindia.net/interviewquestions/jakartastrutsinterviewquestions.shtml##) validator-rules.xml and validation.xml. The validator-rules.xml defines the standard validation routines, these are reusable and used in validation.xml. to define the form specific validations. The validation.xml defines the validations applied to a form bean.

Q. How you will display validation fail errors on jsp page?  
A: Following tag displays all the errors:  
<html:errors/>

Q. How you will enable front-end validation based on the xml in validation.xml?  
A: The <html:javascript> tag to allow front-end validation based on the xml in validation.xml. For  example the code: <html:javascript formName="logonForm" dynamicJavascript="true" staticJavascript="true" /> generates the client side java script for the form "logonForm" as defined in the validation.xml file. The <html:javascript> when added in the jsp file generates the client site validation script.

**What is the use of preparedstatement?** - Preparedstatements are precompiled statements. It is mainly used to speed up the process of inserting/updating/deleting especially when there is a bulk processing.

**What is callable statement? Tell me the way to get the callable statement?** - Callablestatements are used to invoke the stored procedures. You can obtain the callablestatement from Connection using the following methods prepareCall(String sql) prepareCall(String sql, int resultSetType, int resultSetConcurrency)

**In a statement, I am executing a batch. What is the result of the execution?** - It returns the int array. The array contains the affected row count in the corresponding index of the SQL.