JAVA INTERVIEW QUESTIONS

What is the difference between an Interface and an Abstract class?

An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation. An abstract class is a class which may have the usual flavors of class members (private, protected, etc.), but has some abstract methods.

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

The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused. A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

Describe synchronization in respect to multithreading.

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchonization, it is possible for one thread to modify a shared variable while another thread is in the process of using or updating same shared variable. This usually leads to significant errors.

Explain different way of using thread?

The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance. then only interface can help.

What are pass by reference and pass by value?

Pass by Reference means the passing the address itself rather than passing the value. Pass by Value means passing a copy of the value to be passed.

What is HashMap and Map?

Map is Interface and Hashmap is class that implements that.

What is the difference between HashMap and HashTable?

The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls. (HashMap allows null values as key and value whereas Hashtable doesn’t allow). HashMap does not guarantee that the order of the map will remain constant over time. HashMap is unsynchronized and Hashtable is synchronized.

What is the difference between Vector and ArrayList?

Vector is synchronized whereas arraylist is not.

What is the difference between a constructor and a method?

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

What is an Iterator?

Some of the collection classes provide traversal of their contents via a java.util.Iterator interface. This interface allows you to walk through a collection of objects, operating on each object in turn. Remember when using Iterators that they contain a snapshot of the collection at the time the Iterator was obtained; generally it is not advisable to modify the collection itself while traversing an Iterator.

State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.

Public: Public class is visible in other packages, field is visible everywhere (class must be public too)

Private: Private variables or methods may be used only by an instance of the same class that declares the variable or method; a private feature may only be accessed by the class that owns the feature.

Protected: Is available to all classes in the same package and also available to all subclasses of the class that owns the protected feature. This access is provided even to subclasses that reside in a different package from the class that owns the protected feature.

Default: What you get by default i.e., without any access modifier (i.e., public private or protected).It means that it is visible to all within a particular package.

What is an abstract class?

Abstract class must be extended/subclassed (to be useful). It serves as a template. A class that is abstract may not be instantiated (i.e., you may not call its constructor), abstract class may contain static data. Any class with an abstract method is automatically abstract itself, and must be declared as such.
A class may be declared abstract even if it has no abstract methods. This prevents it from being instantiated.

What is static in java?

Static means one per class, not one for each object no matter how many instance of a class might exist. This means that you can use them without creating an instance of a class. Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object. A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final. However, you can't override a static method with a nonstatic method. In other words, you can't change a static method into an instance method in a subclass.

What is final?

A final class can't be extended i.e..; final class may not be subclassed. A final method can't be overridden when its class is inherited. You can't change value of a final variable (is a constant).

What if the main method is declared as private?

The program compiles properly but at runtime it will give "Main method not public." message.

What if the static modifier is removed from the signature of the main method?

Program compiles. But at runtime throws an error "NoSuchMethodError".

What if I write static public void instead of public static void?

Program compiles and runs properly.

What if I do not provide the String array as the argument to the method?

Program compiles but throws a runtime error "NoSuchMethodError".

What is the first argument of the String array in main method?

The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name.

If I do not provide any arguments on the command line, then the String array of Main method will be empty or null?

It is empty. But not null.

How can one prove that the array is not null but empty using one line of code?

Print args.length. It will print 0. That means it is empty. But if it would have been null then it would have thrown a NullPointerException on attempting to print args.length.

What environment variables do I need to set on my machine in order to be able to run Java programs?

CLASSPATH and PATH are the two variables.

Can an application have multiple classes having main method?

Yes it is possible. While starting the application we mention the class name to be run. The JVM will look for the Main method only in the class whose name you have mentioned. Hence there is not conflict amongst the multiple classes having main method.

Can I have multiple main methods in the same class?

No the program fails to compile. The compiler says that the main method is already defined in the class.

Do I need to import java.lang package any time? Why?

No. It is by default loaded internally by the JVM.

Can I import same package/class twice? Will the JVM load the package twice at runtime?

One can import the same package or same class multiple times. Neither compiler nor JVM complains about it. And the JVM will internally load the class only once no matter how many times you import the same class.

What are Checked and Unchecked Exception?

A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses. Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown. e.g., IOException thrown by java.io.FileInputStream's read() method· Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked. With an unchecked exception, however, the compiler doesn't force client programmers either to catch the exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown. e.g., StringIndexOutOfBoundsException thrown by String's charAt() method· Checked exceptions must be caught at compile time. Runtime exceptions do not need to be. Errors often cannot be.

What is overriding?

When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass. When the method is invoked for an object of the class, it is

the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.

What are different types of inner classes?

Nested top-level classes: If you declare a class within a class and specify the static modifier, the compiler treats the class just like any other top-level class. Any class outside the declaring class accesses the nested class with the declaring class name acting similarly to a package. e.g., outer. inner. Top-level inner classes implicitly have access only to static variables. There can also be inner interfaces. All of these are of the nested top-level variety.

Member classes: Member inner classes are just like other member methods and member variables and access to the member class is restricted, just like methods and variables. This means public member class acts similarly to a nested top-level class. The primary difference between member classes and nested top-level classes is that member classes have access to the specific instance of the enclosing class.

Local classes: Local classes are like local variables, specific to a block of code. Their visibility is only within the block of their declaration. In order for the class to be useful beyond the declaration block, it would need to implement a more publicly available interface. Because local classes are not members, the modifiers public, protected, private, and static are not usable.

Anonymous classes: Anonymous inner classes extend local inner classes one level further. As anonymous classes have no name, you cannot provide a constructor.

Are the imports checked for validity at compile time? e.g. will the code containing an import such as java.lang.ABCD compile?

Yes the imports are checked for the semantic validity at compile time. The code containing above line of import will not compile. It will throw an error saying, can not resolve symbol, symbol: class ABCD location: package io import java.io.ABCD;

Does importing a package imports the subpackages as well? e.g. Does importing com.MyTest.* also import com.MyTest.UnitTests.*?

No you will have to import the subpackages explicitly. Importing com.MyTest.* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

What is the difference between declaring a variable and defining a variable?

In declaration we just mention the type of the variable and it's name. We do not initialize it. But defining means declaration + initialization. e.g. String s; is just a declaration while String s = new String ("abcd"); Or String s = "abcd"; are both definitions.

What is the default value of an object reference declared as an instance variable?

null unless we define it explicitly.

Can a top level class be private or protected?

No. A top level class can not be private or protected. It can have either "public" or no modifier. If it does not have a modifier it is supposed to have a default access. If a top level class is declared as private the compiler will complain that the "modifier private is not allowed here". This means that a top level class can not be private. Same is the case with protected.

What type of parameter passing does Java support?

In Java the arguments are always passed by value .

Primitive data types are passed by reference or pass by value?

Primitive data types are passed by value.

Objects are passed by value or by reference?

Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object .

What is serialization?

Serialization is a mechanism by which you can save the state of an object by converting it to a byte stream.

How do I serialize an object to a file?

The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a fileoutputstream. This will save the object to a file.

Which methods of Serializable interface should I implement?

The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods.

How can I customize the serialization process? i.e. how can one have a control over the serialization process?

Yes it is possible to have control over serialization process. The class should implement Externalizable interface. This interface contains two methods namely readExternal and writeExternal. You should implement these methods and write the logic for customizing the serialization process.

What is the common usage of serialization?

Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serialized.

What is Externalizable interface?

Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism. Thus if your class implements this interface, you can customize the serialization process by implementing these methods.

When you serialize an object, what happens to the object references included in the object?

The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized along with the original object.

What one should take care of while serializing the object?

One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

What happens to the static fields of a class during serialization?

There are three exceptions in which serialization doesn't necessarily read and write to the stream. These are

- Serialization ignores static fields, because they are not part of any particular state
- Base class fields are only handled if the base class itself is serializable
- Transient fields.

Does Java provide any construct to find out the size of an object?

No there is not sizeof operator in Java. So there is not direct way to determine the size of an object directly in Java.

Give a simplest way to find out the time a method takes for execution without using any profiling tool?

Read the system time just before the method is invoked and immediately after method returns. Take the time difference, which will give you the time taken by a method for execution.

To put it in code...

```
long start = System.currentTimeMillis ();
//method ();
long end = System.currentTimeMillis ();
System.out.println ("Time taken for execution is " + (end - start));
```

Remember that if the time taken for execution is too small, it might show that it is taking zero milliseconds for execution. Try it on a method which is big enough, in the sense the one which is doing considerable amount of processing.

What are wrapper classes?

Java provides specialized classes corresponding to each of the primitive data types. These are called wrapper classes. They are e.g. Integer, Character, and Double etc.

Why do we need wrapper classes?

It is sometimes easier to deal with primitives as objects. Moreover most of the collection classes store objects and not primitive data types. And also the wrapper classes provide many utility methods also. Because of these reasons we need wrapper classes. And since we create instances of these classes we can store them in any of the collection classes and pass them around as a collection. Also we can pass them around as method parameters where a method expects an object.

What are checked exceptions?

Checked exception is those which the Java compiler forces you to catch. e.g. IOException are checked Exceptions.

What are runtime exceptions?

Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic etc. These are not checked by the compiler at compile time.

What is the difference between error and an exception?

An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. These JVM errors and you can not repair them at runtime. While exceptions are conditions that occur because of bad input etc. e.g. FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference. In most of the

cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.).

How to create custom exceptions?

Your class should extend class Exception, or some more specific type thereof.

If I want an object of my class to be thrown as an exception object, what should I do?

The class should extend from Exception class. Or you can extend your class from some more precise exception type also.

If my class already extends from some other class what should I do if I want an instance of my class to be thrown as an exception object?

One can not do anything in this scenario. Because Java does not allow multiple inheritance and does not provide any exception interface as well.

How does an exception permeate through the code?

An unhandled exception moves up the method stack in search of a matching When an exception is thrown from a code which is wrapped in a try block followed by one or more catch blocks, a search is made for matching catch block. If a matching type is found then that block will be invoked. If a matching type is not found then the exception moves up the method stack and reaches the caller method. Same procedure is repeated if the caller method is included in a try catch block. This process continues until a catch block handling the appropriate type of exception is found. If it does not find such a block then finally the program terminates.

What are the different ways to handle exceptions?

There are two ways to handle exceptions,

- By wrapping the desired code in a try block followed by a catch block to catch the exceptions
- List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions.

What is the basic difference between the 2 approaches to exception handling; try catch block and specifying the candidate exceptions in the throws clause? When should you use which approach?

In the first approach as a programmer of the method, you urself are dealing with the exception. This is fine if you are in a best position to decide should be done in case of an exception. Whereas if it is not the responsibility of the method to deal with it's own exceptions, then do not use this approach. In this case use the second approach. In the second approach we are forcing the caller of the method to catch

the exceptions that the method is likely to throw. This is often the approach library creator's use. They list the exception in the throws clause and we must catch them. You will find the same approach throughout the java libraries we use.

Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

If I write return at the end of the try block, will the finally block still execute?

Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.

If I write System.exit (0); at the end of the try block, will the finally block still execute?

No in this case the finally block will not execute because when you say System.exit (0); the control immediately goes out of the program, and thus finally never executes.

How are Observer and Observable used?

Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

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.

How does Java handle integer overflows and underflows?

It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

Does garbage collection guarantee that a program will not run out of memory?

Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they

are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection.

What is the difference between preemptive scheduling and time slicing?

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

When a thread is created and started, what is its initial state?

A thread is in the ready state after it has been created and started.

What is the purpose of finalization?

The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected.

What is the Locale class?

The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region.

What is the difference between a while statement and a do statement?

A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

What is the difference between static and non-static variables?

A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

How are this() and super() used with constructors?

This() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

What are synchronized methods and synchronized statements?

Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

What are 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.

Can applets communicate with each other?

At this point in time applets may communicate with other applets running in the same virtual machine. If the applets are of the same class, they can communicate via shared static variables. If the applets are of different classes, then each will need a reference to the same class with static variables. In any case the basic idea is to pass the information back and forth through a static variable.

An applet can also get references to all other applets on the same page using the getApplets() method of java.applet.AppletContext. Once you get the reference to an applet, you can communicate with it by using its public members.

It is conceivable to have applets in different virtual machines that talk to a server somewhere on the Internet and store any data that needs to be serialized there. Then, when another applet needs this data, it could connect to this same server. Implementing this is non-trivial.

What are the steps in the JDBC connection?

While making a JDBC connection we go through the following steps :

Step 1: Register the database driver by using :

Class.forName(\" driver class for that specific database\");

Step 2: Now create a database connection using :

Connection con = DriverManager.getConnection(url,username,password);

Step 3: Now Create a query using :

Statement stmt = Connection.Statement(\"select * from TABLE NAME\");

Step 4: Execute the query :

stmt.exceuteUpdate();

How does a try statement determine which catch clause should be used to handle an exception?

When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

Can an unreachable object become reachable again?

An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects.

What method must be implemented by all threads?

All tasks must implement the run() method, whether they are a subclass of Thread or implement the Runnable interface.

What are synchronized methods and synchronized statements?

Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

What is Externalizable?

Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInput in)

What modifiers are allowed for methods in an Interface?

Only public and abstract modifiers are allowed for methods in interfaces.

What are some alternatives to inheritance?

Delegation is an alternative to inheritance. Delegation means that you include an instance of another class as an instance variable, and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn't force you to accept all the methods of the super class: you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to re-use (because it is not a subclass).

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.

What is the difference between preemptive scheduling and time slicing?

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

What is the catch or declare rule for method declarations?

If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

Is Empty .java file a valid source file?

Yes, an empty .java file is a perfectly valid source file.

Can a .java file contain more than one java classes?

Yes, a .java file contains more than one java classes, provided at the most one of them is a public class.

Is String a primitive data type in Java?

No, String is not a primitive data type in Java, even though it is one of the most extensively used object. Strings in Java are instances of String class defined in java.lang package.

Is main a keyword in Java?

No, main is not a keyword in Java.

Is next a keyword in Java?

No, next is not a keyword.

Is delete a keyword in Java?

No, delete is not a keyword in Java. Java does not make use of explicit destructors the way C++ does.

Is exit a keyword in Java?

No. To exit a program explicitly you use exit method in System object.

What happens if you don't initialize an instance variable of any of the primitive types in Java?

Java by default initializes it to the default value for that primitive type. Thus an int will be initialized to 0, a boolean will be initialized to false.

What will be the initial value of an object reference which is defined as an instance variable?

The object references are all initialized to null in Java. However in order to do anything useful with these references, you must set them to a valid object, else you will get NullPointerExceptions everywhere you try to use such default initialized references.

What are the different scopes for Java variables?

The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point in time.

Instance: These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object accessible.

Local: These are the variables that are defined within a method. They remain accessbile only during the course of method execution. When the method finishes execution, these variables fall out of scope.

Static: These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance.

What is the default value of the local variables?

The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the java compiler will not compile the code. It will complain about the local variable not being initialized..

How many objects are created in the following piece of code?

```
MyClass c1, c2, c3;
c1 = new MyClass ();
c3 = new MyClass ();
```

Only 2 objects are created, c1 and c3. The reference c2 is only declared and not initialized.

Can a public class MyClass be defined in a source file named YourClass.java?

No the source file name, if it contains a public class, must be the same as the public class name itself with a .java extension.

Can main method be declared final?

Yes, the main method can be declared final, in addition to being public static.

What will be the output of the following statement? System.out.println ("1" + 3);

It will print 13.

What will be the default values of all the elements of an array defined as an instance variable?

If the array is an array of primitive types, then all the elements of the array will be initialized to the default value corresponding to that primitive type. e.g. All the elements of an array of int will be initialized to 0, while that of boolean type will be initialized to false. Whereas if the array is an array of references (of any type), all the elements will be initialized to null.

What are the Collections API?

The Collections API is a set of classes and interfaces that support operations on collections of objects.

What is the List interface?

The List interface provides support for ordered collections of objects.

What is the Vector class?

The Vector class provides the capability to implement a growable array of objects.

What is an Iterator interface?

The Iterator interface is used to step through the elements of a Collection .

Which java.util classes and interfaces support event handling?

The EventObject class and the EventListener interface support event processing.

What is the GregorianCalendar class?

The GregorianCalendar provides support for traditional Western calendars

What is the Locale class?

The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region .

What is the SimpleTimeZone class?

The SimpleTimeZone class provides support for a Gregorian calendar .

What is the Map interface?

The Map interface replaces the JDK 1.1 Dictionary class and is used associate keys with values.

What is the highest-level event class of the event-delegation model?

The java.util.EventObject class is the highest-level class in the event-delegation class hierarchy.

What is the Collection interface?

The Collection interface provides support for the implementation of a mathematical bag - an unordered collection of objects that may contain duplicates.

What is the Set interface?

The Set interface provides methods for accessing the elements of a finite mathematical set. Sets do not allow duplicate elements.

What is the typical use of Hashtable?

Whenever a program wants to store a key value pair, one can use Hashtable.

I am trying to store an object using a key in a Hashtable. And some other object already exists in that location, then what will happen? The existing object will be overwritten? Or the new object will be stored elsewhere?

The existing object will be overwritten and thus it will be lost.

What is the difference between the size and capacity of a Vector?

The size is the number of elements actually stored in the vector, while capacity is the maximum number of elements it can store at a given instance of time.

Can a vector contain heterogenous objects?

Yes a Vector can contain heterogenous objects. Because a Vector stores everything in terms of Object.

Can an ArrayList contain heterogenous objects?

Yes a ArrayList can contain heterogenous objects. Because a ArrayList stores everything in terms of Object.

What is an enumeration?

An enumeration is an interface containing methods for accessing the underlying data structure from which the enumeration is obtained. It is a construct which collection classes return when you request a collection of all the objects stored in the collection. It allows sequential access to all the elements stored in the collection.

Considering the basic properties of Vector and ArrayList, where will you use Vector and where will you use ArrayList?

The basic difference between a Vector and an ArrayList is that, vector is synchronized while ArrayList is not. Thus whenever there is a possibility of multiple

threads accessing the same instance, one should use Vector. While if not multiple threads are going to access the same instance then use ArrayList. Non synchronized data structure will give better performance than the synchronized one.

Can a vector contain heterogenous objects?

Yes a Vector can contain heterogenous objects. Because a Vector stores everything in terms of Object.

Why threads block or enter to waiting state on I/O?

Threads enters to waiting state or block on I/O because other threads can execute while the I/O operations are performed.

What are transient variables in java?

Transient variables are variable that cannot be serialized.

How Observer and Observable are used?

Subclass of Observable class maintains a list of observers. Whenever an Observable object is updated, it invokes the update() method of each of its observers to notify the observers that it has a changed state. An observer is any object that implements the interface Observer.

What is synchronization?

Synchronization is the ability to control the access of multiple threads to shared resources. Synchronization stops multithreading. With synchronization , at a time only one thread will be able to access a shared resource.

What is List interface?

A List is an ordered collection of objects.

What is a Vector?

A Vector is a grow able array of objects.

What is the difference between yield() and sleep()?

When a object invokes yield() it returns to ready state. But when an object invokes sleep() method enters to not ready state.

What are Wrapper Classes?

They are wrappers to primitive data types. They allow us to access primitives as objects.

Can we call finalize() method ?

Yes. Nobody will stop us to call any method , if it is accessible in our class. But a garbage collector cannot call an object's finalize method if that object is reachable.

What is the difference between time slicing and preemptive scheduling ?

In preemptive scheduling, highest priority task continues execution till it enters a not running state or a higher priority task comes into existence. In time slicing, the task continues its execution for a predefined period of time and reenters the pool of ready tasks.

What is the initial state of a thread when it is created and started?

The thread is in ready state.

Can we declare an anonymous class as both extending a class and implementing an interface?

No. An anonymous class can extend a class or implement an interface, but it cannot be declared to do both

What are the differences between boolean & operator and & operator?

When an expression containing the & operator is evaluated, both operands are evaluated. And the & operator is applied to the operand. When an expression containing && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then only the second operand is evaluated otherwise the second part will not get executed. && is also called short cut and.

What is the use of the finally block?

Finally is the block of code that executes always. The code in finally block will execute even if an exception is occurred. finally will not execute when the user calls System.exit().

What is an abstract method?

An abstract method is a method that doesn't have a body. It is declared with modifier abstract.

What is JDBC?

JDBC technology is an API (included in both J2SE and J2EE releases) that provides cross-DBMS connectivity to a wide range of SQL databases and access to other tabular data sources, such as spreadsheets or flat files. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment

What are stored procedures?

A stored procedure is a set of statements/commands which reside in the database. The stored procedure is precompiled. Each Database has it's own stored procedure language,

What is JDBC Driver?

The JDBC Driver provides vendor-specific implementations of the abstract classes provided by the JDBC API. This driver is used to connect to the database.

What are the steps required to execute a query in JDBC?

First we need to create an instance of a JDBC driver or load JDBC drivers, then we need to register this driver with DriverManager class. Then we can open a connection. By using this connection , we can create a statement object and this object will help us to execute the query.

What is DriverManager?

DriverManager is a class in java.sql package. It is the basic service for managing a set of JDBC drivers.

What is a ResultSet?

A table of data representing a database result set, which is usually generated by executing a statement that queries the database. A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set.

What is Connection?

Connection class represents a connection (session) with a specific database. SQL statements are executed and results are returned within the context of a connection. A Connection object's database is able to provide information describing its tables, its supported SQL grammar, its stored procedures, the capabilities of this connection, and so on. This information is obtained with the getMetaData method.

What does Class.forName return?

A class as loaded by the classloader.

What is Connection pooling?

Connection pooling is a technique used for sharing server resources among requesting clients. Connection pooling increases the performance of Web applications by reusing active database connections instead of creating a new connection with every request. Connection pool manager maintains a pool of open database connections.

What are the different JDB drivers available?

There are mainly four types of JDBC drivers available. They are:

Type 1: JDBC-ODBC Bridge Driver - A JDBC-ODBC bridge provides JDBC API access via one or more ODBC drivers. Note that some ODBC native code and in many cases native database client code must be loaded on each client machine that uses this type of driver. Hence, this kind of driver is generally most appropriate when automatic installation and downloading of a Java technology application is not important. For information on the JDBC-ODBC bridge driver provided by Sun.

Type 2: Native API Partly Java Driver - A native-API partly Java technology-enabled driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

Type 3: Network protocol Driver - A net-protocol fully Java technology-enabled driver translates JDBC API calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect all of its Java technology-based clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC API alternative. It is likely that all vendors of this solution will provide products suitable for Intranet use. In order for these products to also support Internet access they must handle the additional requirements for security, access through firewalls, etc., that the Web imposes. Several vendors are adding JDBC technology-based drivers to their existing database middleware products.

Type 4: JDBC Net pure Java Driver - A native-protocol fully Java technology-enabled driver converts JDBC technology calls into the network protocol used by DBMS's directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Intranet access. Since many of these protocols are proprietary the database vendors themselves will be the primary source for this style of driver. Several database vendors have these in progress.

What is the fastest type of JDBC driver?

Type 4 (JDBC Net pure Java Driver) is the fastest JDBC driver. Type 1 and Type 3 drivers will be slower than Type 2 drivers (the database calls are make at least three translations versus two), and Type 4 drivers are the fastest (only one translation).

Is the JDBC-ODBC Bridge multi-threaded?

No. The JDBC-ODBC Bridge does not support multi threading. The JDBC-ODBC Bridge uses synchronized methods to serialize all of the calls that it makes to

ODBC. Multi-threaded Java programs may use the Bridge, but they won't get the advantages of multi-threading.

What is cold backup, hot backup, warm backup recovery?

Cold backup means all these files must be backed up at the same time, before the database is restarted. Hot backup (official name is 'online backup') is a backup taken of each tablespace while the database is running and is being accessed by the users

What is the advantage of denormalization?

Data denormalization is reverse procedure, carried out purely for reasons of improving performance. It maybe efficient for a high-throughput system to replicate data for certain data.

How do you handle your own transaction?

Connection Object has a method called setAutocommit (boolean flag) . For handling our own transaction we can set the parameter to false and begin your transaction . Finally commit the transaction by calling the commit method.

Describe what happens when an object is created in Java?

Several things happen in a particular order to ensure the object is constructed properly:

- Memory is allocated from heap to hold all instance variables and implementation-specific data of the object and its superclasses. Implementation-specific data includes pointers to class and method data.
- The instance variables of the objects are initialized to their default values.
- The constructor for the most derived class is invoked. The first thing a constructor does is call the constructor for its uppercase. This process continues until the constructor for java.lang.Object is called, as java.lang.Object is the base class for all objects in java.
- Before the body of the constructor is executed, all instance variable initializers and initialization blocks are executed. Then the body of the constructor is executed. Thus, the constructor for the base class completes first and constructor for the most derived class completes last.

In Java, you can create a String object as below : String str = "abc"; & String str = new String("abc"); Why can't a button object be created as : Button bt = "abc"? Why is it compulsory to create a button object as: Button bt = new Button("abc"); Why this is not compulsory in String's case?

Button bt1= "abc"; It is because "abc" is a literal string (something slightly different than a String object, by-the-way) and bt1 is a Button object. That simple. The only object in Java that can be assigned a literal String is java.lang.String. Important to not that you are NOT calling a java.lang.String constuctor when you type String s = "abc"; for example String x = "abc"; String y = "abc"; refer to the same object. While String x1 = new String("abc"); String x2 = new String("abc"); refer to two different objects.

What is the advantage of OOP?

You will get varying answers to this question depending on whom you ask. Major advantages of OOP are:

- **Simplicity:** software objects model real world objects, so the complexity is reduced and the program structure is very clear;
- **Modularity:** each object forms a separate entity whose internal workings are decoupled from other parts of the system;
- **Modifiability:** it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods;
- **Extensibility:** adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones
- **Maintainability:** objects can be maintained separately, making locating and fixing problems easier
- **Re-usability:** objects can be reused in different programs

What are the main differences between Java and C++?

Everything is an object in Java(Single root hierarchy as everything gets derived from java.lang.Object). Java does not have all the complicated aspects of C++ (For ex: Pointers, templates, unions, operator overloading, structures etc..) The Java language promoters initially said "No pointers!", but when many programmers questioned how you can work without pointers, the promoters began saying "Restricted pointers." You can make up your mind whether it's really a pointer or not. In any event, there's no pointer arithmetic. There are no destructors in Java. (automatic garbage collection), Java does not support conditional compile (#ifdef/#ifndef type). Thread support is built into java but not in C++. Java does not support default arguments. There's no scope resolution operator :: in Java. Java

uses the dot for everything, but can get away with it since you can define elements only within a class. Even the method definitions must always occur within a class, so there is no need for scope resolution there either. There's no "goto " statement in Java. Java doesn't provide multiple inheritance (MI), at least not in the same sense that C++ does. Exception handling in Java is different because there are no destructors. Java has method overloading, but no operator overloading. The String class does use the + and += operators to concatenate strings and String expressions use automatic type conversion, but that's a special built-in case. Java is interpreted for the most part and hence platform independent

What are interfaces?

Interfaces provide more sophisticated ways to organize and control the objects in your system. The interface keyword takes the abstract concept one step further. You could think of it as a "pure" abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields, but The interface keyword takes the abstract concept one step further. You could think of it as a "pure" abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields, but an interface says: "This is what all classes that implement this particular interface will look like." Thus, any code that uses a particular interface knows what methods might be called for that interface, and that's all. So the interface is used to establish a "protocol" between classes. (Some object-oriented programming languages have a keyword called protocol to do the same thing.) Typical example from "Thinking in Java":

```
import java.util.*;
interface Instrument {
        int i = 5; // static & final
        // Cannot have method definitions:
        void play(); // Automatically public
        String what();
        void adjust();
}
class Wind implements Instrument {
        public void play() {
                System.out.println("Wind.play()"); }
```

```
        public String what() { return "Wind"; }
        public void adjust() {}
}
```

How can you achieve Multiple Inheritances in Java?

Java's interface mechanism can be used to implement multiple inheritance, with one important difference from c++ way of doing MI: the inherited interfaces must be abstract. This obviates the need to choose between different implementations, as with interfaces there are no implementations.

```
interface CanFight {
void fight();
interface CanSwim {
void swim();
interface CanFly {
void fly();
class ActionCharacter {
        public void fight() {}
        class Hero extends ActionCharacter implements CanFight, CanSwim, CanFly {
        public void swim() {}
        public void fly() {}

}
```

You can even achieve a form of multiple inheritance where you can use the *functionality* of classes rather than just the interface:

```
interface A {
void methodA();
}
class AImpl implements A {
void methodA() { //do stuff }
}
interface B {
void methodB();
}
class BImpl implements B {
void methodB() { //do stuff }
```

```
}
class Multiple implements A, B {
private A a = new A();
private B b = new B();
void methodA() { a.methodA(); }
void methodB() { b.methodB(); }
}
```

This completely solves the traditional problems of multiple inheritance in C++ where name clashes occur between multiple base classes. The coder of the derived class will have to explicitly resolve any clashes. Don't you hate people who point out minor typos? Everything in the previous example is correct, except you need to instantiate an AImpl and BImpl. So class Multiple would look like this:

```
class Multiple implements A, B {
private A a = new AImpl();
private B b = new BImpl();
void methodA() { a.methodA(); }
void methodB() { b.methodB(); }
}
```

What is the difference between StringBuffer and String class?

A string buffer implements a mutable sequence of characters. A string buffer is like a String, but can be modified. At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls. The String class represents character strings. All string literals in Java programs, such as "abc" are constant and implemented as instances of this class; their values cannot be changed after they are created. Strings in Java are known to be immutable. What it means is that every time you need to make a change to a String variable, behind the scene, a "new" String is actually being created by the JVM. For an example: if you change your String variable 2 times, then you end up with 3 Strings: one current and 2 that are ready for garbage collection. The garbage collection cycle is quite unpredictable and these additional unwanted Strings will take up memory until that cycle occurs. For better performance, use StringBuffers for string-type data that will be reused or changed frequently. There is more overhead per class than using String, but you will end up with less overall classes and consequently consume less memory. Describe, in general, how java's

garbage collector works? The Java runtime environment deletes objects when it determines that they are no longer being used. This process is known as garbage collection. The Java runtime environment supports a garbage collector that periodically frees the memory used by objects that are no longer needed. The Java garbage collector is a mark-sweep garbage collector that scans Java's dynamic memory areas for objects, marking those that are referenced. After all possible paths to objects are investigated, those objects that are not marked (i.e. are not referenced) are known to be garbage and are collected. (A more complete description of our garbage collection algorithm might be "A compacting, mark-sweep collector with some conservative scanning".) The garbage collector runs synchronously when the system runs out of memory, or in response to a request from a Java program. Your Java program can ask the garbage collector to run at any time by calling System.gc(). The garbage collector requires about 20 milliseconds to complete its task so, your program should only run the garbage collector when there will be no performance impact and the program anticipates an idle period long enough for the garbage collector to finish its job. Note: Asking the garbage collection to run does not guarantee that your objects will be garbage collected. The Java garbage collector runs asynchronously when the system is idle on systems that allow the Java runtime to note when a thread has begun and to interrupt another thread (such as Windows 95). As soon as another thread becomes active, the garbage collector is asked to get to a consistent state and then terminate.

What is the difference between '==' and equals method?

equals checks for the content of the string objects while == checks for the fact that the two String objects point to same memory location i.e. they are same references.

What are abstract classes, abstract methods?

Simply speaking a class or a method qualified with "abstract" keyword is an abstract class or abstract method. You create an abstract class when you want to manipulate a set of classes through a common interface. All derived-class methods that match the signature of the base-class declaration will be called using the dynamic binding mechanism. If you have an abstract class, objects of that class almost always have no meaning. That is, abstract class is meant to express only the interface and sometimes some default method implementations, and not a particular implementation, so creating an abstract class object makes no sense and are not allowed (compile will give you an error message if you try to create one). An

abstract method is an incomplete method. It has only a declaration and no method body. Here is the syntax for an abstract method declaration: abstract void f(); If a class contains one or more abstract methods, the class must be qualified an abstract. (Otherwise, the compiler gives you an error message.). It's possible to create a class as abstract without including any abstract methods. This is useful when you've got a class in which it doesn't make sense to have any abstract methods, and yet you want to prevent any instances of that class. Abstract classes and methods are created because they make the abstractness of a class explicit, and tell both the user and the compiler how it was intended to be used. For example:

```
abstract class Instrument {
int i; // storage allocated for each
public abstract void play();
public String what() {
return "Instrument";
public abstract void adjust();
}
class Wind extends Instrument {
public void play() {
System.out.println("Wind.play()");
}
public String what() { return "Wind"; }
public void adjust() {}
```

Abstract classes are classes for which there can be no instances at run time. i.e. the implementation of the abstract classes are not complete. Abstract methods are methods which have no definition. i.e. abstract methods have to be implemented in one of the sub classes or else that class will also become Abstract.

What is the difference between an Applet and an Application?

A Java application is made up of a main() method declared as public static void that accepts a string array argument, along with any other classes that main() calls. It lives in the environment that the host OS provides. A Java applet is made up of at least one public class that has to be subclassed from java.awt.Applet. The applet is confined to living in the user's Web browser, and the browser's security rules, (or Sun's appletviewer, which has fewer restrictions). The differences between an applet and an application are as follows:

- Applets can be embedded in HTML pages and downloaded over the Internet whereas Applications have no special support in HTML for embedding or downloading.
- Applets can only be executed inside a java compatible container, such as a browser or appletviewer whereas Applications are executed at command line by java.exe or jview.exe.
- Applets execute under strict security limitations that disallow certain operations (sandbox model security) whereas Applications have no inherent security restrictions.
- Applets don't have the main() method as in applications. Instead they operate on an entirely different mechanism where they are initialized by init(),started by start(),stopped by stop() or destroyed by destroy().

Java says "write once, run anywhere". What are some ways this isn't quite true?

As long as all implementations of java are certified by sun as 100% pure java this promise of "Write once, Run everywhere" will hold true. But as soon as various java core implementations start digressing from each other, this won't be true anymore. A recent example of a questionable business tactic is the surreptitious behavior and interface modification of some of Java's core classes in their own implementation of Java. Programmers who do not recognize these undocumented changes can build their applications expecting them to run anywhere that Java can be found, only to discover that their code works only on Microsoft's own Virtual Machine, which is only available on Microsoft's own operating systems.

What is the difference between a Vector and an Array. Discuss the advantages and disadvantages of both?

Vector can contain objects of different types whereas array can contain objects only of a single type. Vector can expand at run-time, while array length is fixed. Vector methods are synchronized while Array methods are not

What are java beans?

JavaBeans is a portable, platform-independent component model written in the Java programming language, developed in collaboration with industry leaders. It enables developers to write reusable components once and run them anywhere — benefiting from the platform-independent power of Java technology. JavaBeans acts as a Bridge between proprietary component models and provides a seamless and powerful means for developers to build components that run in ActiveX container

applications. JavaBeans are usual Java classes which adhere to certain coding conventions:

- Implements java.io.Serializable interface
- Provides no argument constructor
- Provides getter and setter methods for accessing it's properties

What is RMI?

RMI stands for Remote Method Invocation. Traditional approaches to executing code on other machines across a network have been confusing as well as tedious and error-prone to implement. The nicest way to think about this problem is that some object happens to live on another machine, and that you can send a message to the remote object and get a result as if the object lived on your local machine. This simplification is exactly what Java Remote Method Invocation (RMI) allows you to do. Above excerpt is from "Thinking in java". For more information refer to any book on Java.

What does the keyword "synchronize" mean in java. When do you use it? What are the disadvantages of synchronization?

Synchronize is used when you want to make your methods thread safe. The disadvantage of synchronize is it will end up in slowing down the program. Also if not handled properly it will end up in dead lock.

What gives java it's "write once and run anywhere" nature?

Java is compiled to be a byte code which is the intermediate language between source code and machine code. This byte code is not platform specific and hence can be fed to any platform. After being fed to the JVM, which is specific to a particular operating system, the code platform specific machine code is generated thus making java platform independent.

What are native methods? How do you use them?

Native methods are methods written in other languages like C, C++, or even assembly language. You can call native methods from Java using JNI. Native methods are used when the implementation of a particular method is present in language other than Java say C, C++. To use the native methods in java we use the keyword native For E.g. public native method_a(). This native keyword is signal to the java compiler that the implementation of this method is in a language other than java. Native methods are used when we realize that it would take up a lot of rework to write that piece of already existing code in other language to Java.

What is JDBC? Describe the steps needed to execute a SQL query using JDBC.

We can connect to databases from java using JDBC. It stands for Java Database Connectivity. Here are the steps:

- Register the jdbc driver with the driver manager
- Establish jdbc connection
- Execute an sql statement
- Process the results
- Close the connection
- Before doing these do import java.sql.*

JDBC is java based API for accessing data from the relational databases. JDBC provides a set of classes and interfaces for doing various database operations. The steps are:

- Register/load the jdbc driver with the driver manager.
- Establish the connection thru DriverManager.getConnection();
- Fire a SQL thru conn.executeStatement();
- Fetch the results in a result set
- Process the results
- Close statement/result set and connection object.

How many different types of JDBC drivers are present? Discuss them.

There are four JDBC driver types.

Type 1: JDBC-ODBC Bridge plus ODBC Driver: The first type of JDBC driver is the JDBC-ODBC Bridge. It is a driver that provides JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to work. This driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.

Type 2: Native-API partly-Java Driver: The Native to API driver converts JDBC commands to DBMS-specific native calls. This is much like the restriction of Type 1 drivers. The client must have some binary code loaded on its machine. These drivers do have an advantage over Type 1 drivers because they interface directly with the database.

Type 3: JDBC-Net Pure Java Driver: The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent

to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.

Type 4: Native-Protocol Pure Java Driver: These are pure Java drivers that communicate directly with the vendor's database. They do this by converting JDBC commands directly into the database engine's native protocol. This driver has no additional translation or middleware layer, which improves performance tremendously.

Discuss about Access specifiers: "public", "protected", "private", nothing?

In the case of Public, Private and Protected, that is used to describe which programs can access that class or method: Public – any other class from any package can instantiate and execute the classes and methods. Protected – only subclasses and classes inside of the package can access the classes and methods. Private – the original class is the only class allowed to executed the methods.

What does the "final" keyword mean in front of a variable? A method? A class?

FINAL for a variable : value is constant

FINAL for a method : cannot be overridden

FINAL for a class : cannot be derived

A final variable cannot be reassigned, but it is not constant. For instance,

final StringBuffer x = new StringBuffer()

x.append("hello");

is valid. X cannot have a new value in it, but nothing stops operations on the object that it refers, including destructive operations. Also, a final method cannot be overridden or hidden by new access specifications. This means that the compiler can choose to in-line the invocation of such a method.(I don't know if any compiler actually does this, but it's true in theory.) The best example of a final class is String, which defines a class that cannot be derived.

Does Java have "goto"?

No.

What synchronization constructs does Java provide? How do they work?

The two common features that are used are:

Synchronized keyword: Used to synchronize a method or a block of code. When you synchronize a method, you are in effect synchronizing the code within the

method using the monitor of the current object for the lock. The following have the same effect.

```
synchronized void foo() {
}
```

and

```
void foo() {
    synchronized(this) { }
}
```

If you synchronize a static method, then you are synchronizing across all objects of the same class, i.e. the monitor you are using for the lock is one per class, not one per object.

wait/notify: wait() needs to be called from within a synchronized block. It will first release the lock acquired from the synchronization and then wait for a signal. In Posix C, this part is equivalent to the pthread_cond_wait method, which waits for an OS signal to continue. When somebody calls notify() on the object, this will signal the code which has been waiting, and the code will continue from that point. If there are several sections of code that are in the wait state, you can call notifyAll() which will notify all threads that are waiting on the monitor for the current object. Remember that both wait() and notify() have to be called from blocks of code that are synchronized on the monitor for the current object.

Does Java have multiple inheritance?

Java does not support multiple inheritance directly but it does thru the concept of interfaces. We can make a class implement a number of interfaces if we want to achieve multiple inheritance type of functionality of C++.

How does exception handling work in Java?

It separates the working/functional code from the error-handling code by way of try-catch clauses.

It allows a clean path for error propagation. If the called method encounters a situation it can't manage, it can throw an exception and let the calling method deal with it.

By enlisting the compiler to ensure that "exceptional" situations are anticipated and accounted for, it enforces powerful coding.

Exceptions are of two types: Compiler-enforced exceptions, or checked exceptions. Runtime exceptions, or unchecked exceptions. Compiler-enforced (checked) exceptions are instances of the Exception class or one of its subclasses —

excluding the RuntimeException branch. The compiler expects all checked exceptions to be appropriately handled. Checked exceptions must be declared in the throws clause of the method throwing them — assuming, of course, they're not being caught within that same method. The calling method must take care of these exceptions by either catching or declaring them in its throws clause. Thus, making an exception checked forces us to pay heed to the possibility of it being thrown. An example of a checked exception is java.io.IOException. As the name suggests, it throws whenever an input/output operation is abnormally terminated.

Does Java have destructors?

Garbage collector does the job working in the background Java does not have destructors; but it has finalizers that does a similar job. The syntax is

public void finalize(){ }

If an object has a finalizer, the method is invoked before the system garbage collects the object

What does the "abstract" keyword mean in front of a method? A class?

Abstract keyword declares either a method or a class. If a method has a abstract keyword in front of it, it is called abstract method. Abstract method has no body. It has only arguments and return type. Abstract methods act as placeholder methods that are implemented in the subclasses. Abstract classes can't be instantiated. If a class is declared as abstract, no objects of that class can be created. If a class contains any abstract method it must be declared as abstract

Are Java constructors inherited ? If not, why not?

You cannot inherit a constructor. That is, you cannot create a instance of a subclass using a constructor of one of it's superclasses. One of the main reasons is because you probably don't want to override the superclasses constructor, which would be possible if they were inherited. By giving the developer the ability to override a superclasses constructor you would erode the encapsulation abilities of the language.

Do I need to use synchronized on setValue(int)?

It depends whether the method affects method local variables, class static or instance variables. If only method local variables are changed, the value is said to be confined by the method and is not prone to threading issues.

What is the SwingUtilities.invokeLater(Runnable) method for?

The static utility method invokeLater(Runnable) is intended to execute a new runnable thread from a Swing application without disturbing the normal sequence of event dispatching from the Graphical User Interface (GUI). The method places the runnable object in the queue of Abstract Windowing Toolkit (AWT) events that are due to be processed and returns immediately. The runnable object's run() method is only called when it reaches the front of the queue. The deferred effect of the invokeLater(Runnable) method ensures that any necessary updates to the user interface can occur immediately, and the runnable work will begin as soon as those high priority events are dealt with. The invoke later method might be used to start work in response to a button click that also requires a significant change to the user interface, perhaps to restrict other activities, while the runnable thread executes.

What is the volatile modifier for?

The volatile modifier is used to identify variables whose values should not be optimized by the Java Virtual Machine, by caching the value for example. The volatile modifier is typically used for variables that may be accessed or modified by numerous independent threads and signifies that the value may change without synchronization.

Which class is the wait() method defined in?

The wait() method is defined in the Object class, which is the ultimate superclass of all others. So the Thread class and any Runnable implementation inherit this method from Object. The wait() method is normally called on an object in a multi-threaded program to allow other threads to run. The method should only be called by a thread that has ownership of the object's monitor, which usually means it is in a synchronized method or statement block.

What is a working thread?

A working thread, more commonly known as a worker thread is the key part of a design pattern that allocates one thread to execute one task. When the task is complete, the thread may return to a thread pool for later use. In this scheme a thread may execute arbitrary tasks, which are passed in the form of a Runnable method argument, typically execute(Runnable). The runnable tasks are usually stored in a queue until a thread host is available to run them. The worker thread design pattern is usually used to handle many concurrent tasks where it is not important which finishes first and no single task needs to be coordinated with another. The task queue controls how many threads run concurrently to improve the overall performance of the system. However, a worker thread framework requires

relatively complex programming to set up, so should not be used where simpler threading techniques can achieve similar results.

What is a green thread?

A green thread refers to a mode of operation for the Java Virtual Machine (JVM) in which all code is executed in a single operating system thread. If the Java program has any concurrent threads, the JVM manages multi-threading internally rather than using other operating system threads. There is a significant processing overhead for the JVM to keep track of thread states and swap between them, so green thread mode has been deprecated and removed from more recent Java implementations. Current JVM implementations make more efficient use of native operating system threads.

What are native operating system threads?

Native operating system threads are those provided by the computer operating system that plays host to a Java application, be it Windows, Mac or GNU/Linux. Operating system threads enable computers to run many programs simultaneously on the same central processing unit (CPU) without clashing over the use of system resources or spending lots of time running one program at the expense of another. Operating system thread management is usually optimised to specific microprocessor architecture and features so that it operates much faster than Java green thread processing.

How could Java classes direct program messages to the system console, but error messages, say to a file?

The class System has a variable out that represents the standard output, and the variable err that represents the standard error device. By default, they both point at the system console. This how the standard output could be re-directed:

```
Stream st = new Stream (new FileOutputStream ("techinterviews_com.txt"));
System.setErr(st);
System.setOut(st);
```

What's the difference between an interface and an abstract class?

An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

Why would you use a synchronized block vs. synchronized method?

Synchronized blocks place locks for shorter periods than synchronized methods.

Explain the usage of the keyword transient?

This keyword indicates that the value of this member variable does not have to be serialized with the object. When the class will be de-serialized, this variable will be initialized with a default value of its data type (i.e. zero for integers).

How can you force garbage collection?

You can't force GC, but could request it by calling System.gc(). JVM does not guarantee that GC will be started immediately.

How do you know if an explicit object casting is needed?

If you assign a superclass object to a variable of a subclass's data type, you need to do explicit casting. For example:

Object a; Customer b; b = (Customer) a;

When you assign a subclass to a variable having a superclass type, the casting is performed automatically.

What's the difference between the methods sleep() and wait()?

The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

Can you write a Java class that could be used both as an applet as well as an application?

Yes. Add a main() method to the applet.

What's the difference between constructors and other methods?

Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.

Can you call one constructor from another if a class has multiple constructors?

Yes. Use this() syntax.

Explain the usage of Java packages.

This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.

If a class is located in a package, what do you need to change in the OS environment to be able to use it?

You need to add a directory or a jar file that contains the package directories to the CLASSPATH environment variable. Let's say a class Employee belongs to a package com.xyz.hr; and is located in the file c:/dev/com.xyz.hr.Employee.java. In this case, you'd need to add c:/dev to the variable CLASSPATH. If this class contains the method main(), you could test it from a command prompt window as follows: c:\>java com.xyz.hr.Employee

What's the difference between J2SDK 1.5 and J2SDK 5.0?

There's no difference, Sun Microsystems just re-branded this version.

What would you use to compare two String variables - the operator == or the method equals()?

I'd use the method equals() to compare the values of the Strings and the = = to check if two variables point at the same instance of a String object.

Does it matter in what order catch statements for FileNotFoundException and IOException are written?

Yes, it does. The FileNotFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

Can an inner class declared inside of a method access local variables of this method?

It's possible if these variables are final.

What can go wrong if you replace && with & in the following code:

```
String a=null;
if (a!=null && a.length()>10)
{...}
```

A single ampersand here would lead to a NullPointerException.

What's the main difference between a Vector and an ArrayList?

Java Vector class is internally synchronized and ArrayList is not.

When should the method invokeLater()be used?

This method is used to ensure that Swing components are updated through the event-dispatching thread.

How can a subclass call a method or a constructor defined in a superclass?

Use the following syntax: super.myMethod(); To call a constructor of the superclass, just write super(); in the first line of the subclass's constructor.

What's the difference between a queue and a stack?

Stacks works by last-in-first-out rule (LIFO), while queues use the FIFO rule.

You can create an abstract class that contains only abstract methods. On the other hand, you can create an interface that declares the same methods. So can you use abstract classes instead of interfaces?

Sometimes. But your class may be a descendent of another class and in this case the interface is your only option.

What comes to mind when you hear about a young generation in Java?

Garbage collection.

What comes to mind when someone mentions a shallow copy in Java?

Object cloning.

If you're overriding the method equals() of an object, which other method you might also consider?

hashCode()

You are planning to do an indexed search in a list of objects. Which of the two Java collections should you use: ArrayList or LinkedList?

ArrayList

How would you make a copy of an entire Java object with its state?

Have this class implement Cloneable interface and call its method clone().

How can you minimize the need of garbage collection and make the memory use more effective?

Use object pooling and weak object references.

There are two classes: A and B. The class B need to inform a class A when some important event has happened. What Java technique would you use to implement it?

If these classes are threads I'd consider notify() or notifyAll(). For regular classes you can use the Observer interface.

What access level do you need to specify in the class declaration to ensure that only classes from the same directory can access it?

You do not need to specify any access level, and Java will use a default package access level.

What is the difference between procedural and object-oriented programs?

- 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.

- In procedural program, data is exposed to the whole program whereas in OOP's 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 UNICODE?

Unicode is used for internal representation of characters and strings and it uses 16 bits to represent each other.

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?

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.

Overloading does not block inheritance from the superclass whereas overriding blocks inheritance from the superclass.

In overloading, separate methods share the same name whereas in overriding, subclass method replaces the superclass.

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 a reflection package?

java. lang. reflect package has the ability to analyze itself in runtime.

What are 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 method and constant 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 is an abstract class?

An abstract class is a class designed with implementation gaps for subclasses to fill in and is deliberately incomplete.

What is the difference between Integer and int?

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.

Integer can be used as an argument for a method that requires an object, whereas int can be used for calculations.

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?

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.

In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.

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?

String objects are constants and immutable whereas StringBuffer objects are not.

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 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 are 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 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(). HasMoreElements() 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.

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 Stream classes are defined by using two abstract classes, namely InputStream and OutputStream. Character Stream 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?

- OBDC is for Microsoft and JDBC is for Java applications.
- ODBC can't be directly used with Java because it uses a C interface.
- ODBC makes use of pointers which have been removed totally from Java.
- 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.
- 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.
- JDBC API is a natural Java interface and is built on ODBC. JDBC retains some of the basic features of ODBC.

What are the types of JDBC Driver Models and explain them?

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?

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.

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");

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");

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?

JDBC-ODBC Bridge driver (is a driver that uses native(C language) libraries and makes calls to an existing ODBC driver to access a database engine).

What are the types of statements in JDBC?

Statement: to be used createStatement() method for executing single SQL statement

PreparedStatement: Use preparedStatement() method for executing same SQL statement over and over

CallableStatement: Use 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 the difference between TCP/IP and UDP?

TCP/IP is a two-way communication between the client and the server and it is a reliable and there is a confirmation regarding reaching the message to the destination. It is like a phone call. UDP is a one-way communication only between the client and the server and it is not a reliable and there is no confirmation regarding reaching the message to the destination. It is like a postal mail.

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 URL?

URL stands for Uniform Resource Locator and it points to resource files on the Internet. URL has four components: http://www. address. com:80/index.html, where http - protocol name, address - IP address or host name, 80 - port number and index.html - file path.

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 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.

Is "abc" a primitive value?

The String literal "abc" is not a primitive value. It is a String object.

What restrictions are placed on the values of each case of a switch statement?

During compilation, the values of each case of a switch statement must evaluate to a value that can be promoted to an int value.

What modifiers may be used with an interface declaration?

An interface may be declared as public or abstract.

Is a class a subclass of itself?

A class is a subclass of itself.

What is the difference between a while statement and a do statement?

A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

What modifiers can be used with a local inner class?

A local inner class may be final or abstract.

What is the purpose of the File class?

The File class is used to create objects that provide access to the files and directories of a local file system.

Can an exception be rethrown?

Yes, an exception can be rethrown.

When does the compiler supply a default constructor for a class?

The compiler supplies a default constructor for a class if no other constructors are provided.

If a method is declared as protected, where may the method is accessed?

A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

Which non-Unicode letter characters may be used as the first character of an identifier?

The non-Unicode letter characters $ and _ may appear as the first character of an identifier

What restrictions are placed on method overloading?

Two methods may not have the same name and argument list but different return types.

What is casting?

There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

What is the return type of a program's main() method?

A program's main() method has a void return type.

What class of exceptions is generated by the Java run-time system?

The Java runtime system generates RuntimeException and Error exceptions.

What class allows you to read objects directly from a stream?

The ObjectInputStream class supports the reading of objects from input streams.

What is the difference between a field variable and a local variable?

A field variable is a variable that is declared as a member of a class. A local variable is a variable that is declared local to a method.

How are this() and super() used with constructors?

this() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

What is the relationship between a method's throws clause and the exceptions that can be thrown during the method's execution?

A method's throws clause must declare any checked exceptions that are not caught within the body of the method.

Why are the methods of the Math class static?

So they can be invoked as if they are a mathematical code library.

What are the legal operands of the instanceof operator?

The left operand is an object reference or null value and the right operand is a class, interface, or array type.

What 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.

If an object is garbage collected, can it become reachable again?

Once an object is garbage collected, it ceases to exist. It can no longer become reachable again.

What are E and PI?

E is the base of the natural logarithm and PI is mathematical value pi.

Are true and false keywords?

The values true and false are not keywords.

What is the difference between the File and RandomAccessFile classes?

The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

What happens when you add a double value to a String?

The result is a String object.

What is your platform's default character encoding?

If you are running Java on English Windows platforms, it is probably Cp1252. If you are running Java on English Solaris platforms, it is most likely 8859_1.

Which package is always imported by default?

The java.lang package is always imported by default.

What interface must an object implement before it can be written to a stream as an object?

An object must implement the Serializable or Externalizable interface before it can be written to a stream as an object.

How can my application get to know when a HttpSession is removed?

Define a Class HttpSessionNotifier which implements HttpSessionBindingListener and implement the functionality what you need in valueUnbound() method. Create an instance of that class and put that instance in HttpSession.

Whats the difference between notify() and notifyAll()?

notify() is used to unblock one waiting thread; notifyAll() is used to unblock all of them. Using notify() is preferable (for efficiency) when only one blocked thread can benefit from the change (for example, when freeing a buffer back into a pool). notifyAll() is necessary (for correctness) if multiple threads should resume (for example, when releasing a "writer" lock on a file might permit all "readers" to resume).

Why can't I say just abs() or sin() instead of Math.abs() and Math.sin()?

The import statement does not bring methods into your local name space. It lets you abbreviate class names, but not get rid of them altogether. That's just the way it works, you'll get used to it. It's really a lot safer this way. However, there is actually a little trick you can use in some cases that gets you what you want. If your top-level class doesn't need to inherit from anything else, make it

inherit from java.lang.Math. That *does* bring all the methods into your local name space. But you can't use this trick in an applet, because you have to inherit from java.awt.Applet. And actually, you can't use it on java.lang.Math at all, because Math is a "final" class which means it can't be extended.

Why are there no global variables in Java?

Global variables are considered bad form for a variety of reasons: Adding state variables breaks referential transparency (you no longer can understand a statement or expression on its own: you need to understand it in the context of the settings of the global variables), State variables lessen the cohesion of a program: you need to know more to understand how something works. A major point of Object-Oriented programming is to break up global state into more easily understood collections of local state, When you add one variable, you limit the use of your program to one instance. What you thought was global, someone else might think of as local: they may want to run two copies of your program at once. For these reasons, Java decided to ban global variables.

What does it mean that a class or member is final?

A final class can no longer be subclassed. Mostly this is done for security reasons with basic classes like String and Integer. It also allows the compiler to make some optimizations, and makes thread safety a little easier to achieve. Methods may be declared final as well. This means they may not be overridden in a subclass. Fields can be declared final, too. However, this has a completely different meaning. A final field cannot be changed after it's initialized, and it must include an initializer statement where it's declared. For example, public final double c = 2.998; It's also possible to make a static field final to get the effect of C++'s const statement or some uses of C's #define, e.g. public static final double c = 2.998;

What does it mean that a method or class is abstract?

An abstract class cannot be instantiated. Only its subclasses can be instantiated. You indicate that a class is abstract with the abstract keyword like this: public abstract class Container extends Component {

Abstract classes may contain abstract methods. A method declared abstract is not actually implemented in the current class. It exists only to be overridden in subclasses. It has no body. For example, public abstract float price();

Abstract methods may only be included in abstract classes. However, an abstract class is not required to have any abstract methods, though most of them

do. Each subclass of an abstract class must override the abstract methods of its superclasses or itself be declared abstract.

What is a transient variable?

transient variable is a variable that may not be serialized.

How are Observer and Observable used?

Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

Can a lock be acquired on a class?

Yes, a lock can be acquired on a class. This lock is acquired on the class's Class object.

What state does a thread enter when it terminates its processing?

When a thread terminates its processing, it enters the dead state.

How does Java handle integer overflows and underflows?

It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

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

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

Is sizeof a keyword?

The sizeof operator is not a keyword.

Does garbage collection guarantee that a program will not run out of memory?

Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection

Can an object's finalize() method be invoked while it is reachable?

An object's finalize() method cannot be invoked by the garbage collector while the object is still reachable. However, an object's finalize() method may be invoked by other objects.

What value does readLine() return when it has reached the end of a file?

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

Can a for statement loop indefinitely?

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

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

The default value of an String type is null.

What is a task's priority and how is it used in scheduling?

A task's priority is an integer value that identifies the relative order in which it should be executed with respect to other tasks. The scheduler attempts to schedule higher priority tasks before lower priority tasks.

What is the range of the short type?

The range of the short type is -(2^15) to 2^15 - 1.

What is the purpose of garbage collection?

The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources may be reclaimed and reused.

What do you understand by private, protected and public?

These are accessibility modifiers. Private is the most restrictive, while public is the least restrictive. There is no real difference between protected and the default type (also known as package protected) within the context of the same package, however the protected keyword allows visibility to a derived class in a different package.

What is Downcasting?

Downcasting is the casting from a general to a more specific type, i.e. casting down the hierarchy

Can a method be overloaded based on different return type but same argument type?

No, because the methods can be called without using their return type in which case there is ambiquity for the compiler

What happens to a static var that is defined within a method of a class ?

Can't do it. You'll get a compilation error

How many static init can you have ?

As many as you want, but the static initializers and class variable initializers are executed in textual order and may not refer to class variables declared in the class whose declarations appear textually after the use, even though these class variables are in scope.

What is the difference amongst JVM Spec, JVM Implementation, and JVM Runtime ?

The JVM spec is the blueprint for the JVM generated and owned by Sun. The JVM implementation is the actual implementation of the spec by a vendor and the JVM runtime is the actual running instance of a JVM implementation

Describe what happens when an object is created in Java?

Several things happen in a particular order to ensure the object is constructed properly: Memory is allocated from heap to hold all instance variables and implementation-specific data of the object and its superclasses. Implemenation-specific data includes pointers to class and method data. The instance variables of the objects are initialized to their default values. The constructor for the most derived class is invoked. The first thing a constructor does is call the consctructor for its superclasses. This process continues until the constrcutor for java.lang.Object is called, as java.lang.Object is the base class for all objects in java. Before the body of the constructor is executed, all instance variable initializers and initialization blocks are executed. Then the body of the constructor is executed. Thus, the constructor for the base class completes first and constructor for the most derived class completes last.

What does the "final" keyword mean in front of a variable? A method? A class?

FINAL for a variable: value is constant. FINAL for a method: cannot be overridden. FINAL for a class: cannot be derived

What is the difference between instanceof and isInstance?

instanceof is used to check to see if an object can be cast into a specified type without throwing a cast class exception. isInstance() Determines if the specified Object is assignment-compatible with the object represented by this Class. This method is the dynamic equivalent of the Java language instanceof operator. The method returns true if the specified Object argument is non-null and can be cast to the reference type represented by this Class object without raising a ClassCastException. It returns false otherwise.

What is the query used to display all tables names in SQL Server (Query analyzer)?

select * from information_schema.tables

I made my class Cloneable but I still get Can't access protected method clone. Why?

Some of the Java books imply that all you have to do in order to have your class support clone() is implement the Cloneable interface. Not so. Perhaps that was the intent at some point, but that's not the way it works currently. As it stands, you have to implement your own public clone() method, even if it doesn't do anything special and just calls super.clone().

Why is XML such an important development?

It removes two constraints which were holding back Web developments: dependence on a single, inflexible document type (HTML) which was being much abused for tasks it was never designed for; the complexity of full SGML, whose syntax allows many powerful but hard-to-program options. XML allows the flexible development of user-defined document types. It provides a robust, non-proprietary, persistent, and verifiable file format for the storage and transmission of text and data both on and off the Web; and it removes the more complex options of SGML, making it easier to program for.

What is the fastest type of JDBC driver?

JDBC driver performance will depend on a number of issues:

- the quality of the driver code,
- the size of the driver code,
- the database server and its load,
- network topology,
- the number of times your request is translated to a different API.

In general, all things being equal, you can assume that the more your request and response change hands, the slower it will be. This means that Type 1 and Type 3 drivers will be slower than Type 2 drivers (the database calls are make at least three translations versus two), and Type 4 drivers are the fastest (only one translation).

How do I find whether a parameter exists in the request object?

```
boolean hasFoo = !(request.getParameter("foo") == null
|| request.getParameter("foo").equals(""));
```

or

```
boolean hasParameter =
request.getParameterMap().contains(theParameter); //(which works in Servlet 2.3+)
```

How can I send user authentication information while makingURLConnection?

You'll want to use HttpURLConnection.setRequestProperty and set all the appropriate headers to HTTP authorization.

What is the purpose of finalization?

The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected.

What is the difference between the Boolean & operator and the && operator?

If an expression involving the Boolean & operator is evaluated, both operands are evaluated. Then the & operator is applied to the operand. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. The && operator is then applied to the first and second operands. If the first operand evaluates to false, the evaluation of the second operand is skipped.

How many times may an object's finalize() method be invoked by the garbage collector?

An object's finalize() method may only be invoked once by the garbage collector.

What is the purpose of the finally clause of a try-catch-finally statement?

The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

What is the argument type of a program's main() method?

A program's main() method takes an argument of the String[] type.

Which Java operator is right associative?

The = operator is right associative.

Can a double value be cast to a byte?

Yes, a double value can be cast to a byte.

What is the difference between a break statement and a continue statement?

A break statement results in the termination of the statement to which it applies (switch, for, do, or while). A continue statement is used to end the current loop iteration and return control to the loop statement.

What must a class do to implement an interface?

It must provide all of the methods in the interface and identify the interface in its implements clause.

What is the advantage of the event-delegation model over the earlier event-inheritance model?

The event-delegation model has two advantages over the event-inheritance model. First, it enables event handling to be handled by objects other than the ones that generate the events (or their containers). This allows a clean separation between a component's design and its use. The other advantage of the event-delegation model is that it performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to repeatedly process unhandled events, as is the case of the event-inheritance model.

How are commas used in the intialization and iteration parts of a for statement?

Commas are used to separate multiple statements within the initialization and iteration parts of a for statement.

What is an abstract method?

An abstract method is a method whose implementation is deferred to a subclass.

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

The read() method returns -1 when it has reached the end of a file.

Can a Byte object be cast to a double value?

No, an object cannot be cast to a primitive value.

What is the difference between a static and a non-static inner class?

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

If a variable is declared as private, where may the variable be accessed?

A private variable may only be accessed within the class in which it is declared.

What is an object's lock and which objects have locks?

An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock. All objects and classes have locks. A class's lock is acquired on the class's Class object.

What is the % operator?

It is referred to as the modulo or remainder operator. It returns the remainder of dividing the first operand by the second operand.

When can an object reference be cast to an interface reference?

An object reference be cast to an interface reference when the object implements the referenced interface.

Which class is extended by all other classes?

The Object class is extended by all other classes.

Can an object be garbage collected while it is still reachable?

A reachable object cannot be garbage collected. Only unreachable objects may be garbage collected.

Is the ternary operator written x : y ? z or x ? y : z ?

It is written x ? y : z.

How is rounding performed under integer division?

The fractional part of the result is truncated. This is known as rounding toward zero.

What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

What classes of exceptions may be caught by a catch clause?

A catch clause can catch any exception that may be assigned to the Throwable type. This includes the Error and Exception types.

If a class is declared without any access modifiers, where may the class be accessed?

A class that is declared without any access modifiers is said to have package access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

Does a class inherit the constructors of its superclass?

A class does not inherit constructors from any of its superclasses.

What is the purpose of the System class?

The purpose of the System class is to provide access to system resources.

Name the eight primitive Java types.

The eight primitive types are byte, char, short, int, long, float, double, and boolean.

Which class should you use to obtain design information about an object?

The Class class is used to obtain information about an object's design.

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

Yes

Can an Interface be final?

No

Can an Interface have an inner class?

Yes.

```
public interface abc
{
        static int i=0; void dd();
        class a1
        {
                a1()
                {
                        int j;
                        System.out.println("inside");
                };
                public static void main(String a1[])
                {
                        System.out.println("in interfia");
                }
        }
}
```

Can we define private and protected modifiers for variables in interfaces?

No

What is Externalizable?

Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInput in)

What modifiers are allowed for methods in an Interface?

Only public and abstract modifiers are allowed for methods in interfaces.

What is a local, member and a class variable?

Variables declared within a method are "local" variables. Variables declared within the class i.e. not within any methods are "member" variables (global

variables). Variables declared within the class i.e. not within any methods and are defined as "static" are class variables

What are the different identifier states of a Thread?

The different identifiers of a Thread are: R - Running or runnable thread, S - Suspended thread, CW - Thread waiting on a condition variable, MW - Thread waiting on a monitor lock, MS - Thread suspended waiting on a monitor lock

What are some alternatives to inheritance?

Delegation is an alternative to inheritance. Delegation means that you include an instance of another class as an instance variable, and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn't force you to accept all the methods of the super class: you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to re-use (because it is not a subclass).

Why isn't there operator overloading?

Because C++ has proven by example that operator overloading makes code almost impossible to maintain. In fact there very nearly wasn't even method overloading in Java, but it was thought that this was too useful for some very basic methods like print(). Note that some of the classes like DataOutputStream have unoverloaded methods like writeInt() and writeByte().

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();

Why do threads block on I/O?

Threads block on i/o (that is enters the waiting state) so that other threads may execute while the I/O operation is performed.

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.

Is null a keyword?

The null value is not a keyword.

Which characters may be used as the second character of an identifier, but not as the first character of an identifier?

The digits 0 through 9 may not be used as the first character of an identifier but they may be used after the first character of an identifier.

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

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

How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters?

Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

What are wrapped classes?

Wrapped classes are classes that allow primitive types to be accessed as objects.

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

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

What is the difference between preemptive scheduling and time slicing?

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

What is a native method?

A native method is a method that is implemented in a language other than Java.

What are order of precedence and associativity, and how are they used?

Order of precedence determines the order in which operators are evaluated in expressions. Associativity determines whether an expression is evaluated left-to-right or right-to-left

What is the catch or declare rule for method declarations?

If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

Can an anonymous class be declared as implementing an interface and extending a class?

An anonymous class may implement an interface or extend a superclass, but may not be declared to do both.

What is the range of the char type?

The range of the char type is 0 to 2^16 - 1.

What is garbage collection? What is the process that is responsible for doing that in java?

Reclaiming the unused memory by the invalid objects. Garbage collector is responsible for this process

What kind of thread is the Garbage collector thread?

It is a daemon thread.

What is a daemon thread?

These are the threads which can run without user intervention. The JVM can exit when there are daemon thread by killing them abruptly.

How will you invoke any external process in Java?

Runtime.getRuntime().exec(….)

What is the finalize method do?

Before the invalid objects get garbage collected, the JVM give the user a chance to clean up some resources before it got garbage collected.

What is mutable object and immutable object?

If a object value is changeable then we can call it as Mutable object. (Ex., StringBuffer, …) If you are not allowed to change the value of an object, it is immutable object. (Ex., String, Integer, Float, …)

What is the basic difference between string and StringBuffer object?

String is an immutable object. StringBuffer is a mutable object.

What is the purpose of Void class?

The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the primitive Java type void.

What is reflection?

Reflection allows programmatic access to information about the fields, methods and constructors of loaded classes, and the use reflected fields, methods, and constructors to operate on their underlying counterparts on objects, within security restrictions.

What is the base class for Error and Exception?

Throwable

What is the byte range?

-128 to 127

What is the implementation of destroy method in java.. is it native or java code?

This method is not implemented.

What is a package?

To group set of classes into a single unit is known as packaging. Packages provide wide namespace ability.

What are the approaches that you will follow for making a program very efficient?

By avoiding too much of static methods avoiding the excessive and unnecessary use of synchronized methods Selection of related classes based on the application (meaning synchronized classes for multiuser and non-synchronized classes for single user) Usage of appropriate design patterns Using cache methodologies for remote invocations Avoiding creation of variables within a loop and lot more.

What is a Database Metadata?

Comprehensive information about the database as a whole.

What is Locale?

A Locale object represents a specific geographical, political, or cultural region

How will you load a specific locale?

Using ResourceBundle.getBundle(…);

What are JIT and its use?

Really, just a very fast compiler… In this incarnation, pretty much a one-pass compiler — no offline computations. So you can’t look at the whole method, rank the

expressions according to which ones are re-used the most, and then generate code. In theory terms, it's an on-line problem.

Is JVM a compiler or an interpreter?

Interpreter

When you think about optimization, what is the best way to find out the time/memory consuming process?

Using profiler

What is the purpose of assert keyword used in JDK1.4.x?

In order to validate certain expressions. It effectively replaces the if block and automatically throws the AssertionError on failure. This keyword should be used for the critical arguments. Meaning, without that the method does nothing.

How will you get the platform dependent values like line separator, path separator, etc., ?

Using Sytem.getProperty(…) (line.separator, path.separator, …)

What is skeleton and stub? what is the purpose of those?

Stub is a client side representation of the server, which takes care of communicating with the remote server. Skeleton is the server side representation. But that is no more in use… it is deprecated long before in JDK.

What is the final keyword denotes?

final keyword denotes that it is the final implementation for that method or variable or class. You can't override that method/variable/class any more.

What is the significance of ListIterator?

You can iterate back and forth.

What is the major difference between LinkedList and ArrayList?

LinkedList are meant for sequential accessing. ArrayList are meant for random accessing.

What is nested class?

If all the methods of a inner class is static then it is a nested class.

What is inner class?

If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.

What is composition?

Holding the reference of the other class within some other class is known as composition.

What is aggregation?

It is a special type of composition. If you expose all the methods of a composite class and route the method call to the composite method through its reference, then it is called aggregation.

What are the methods in Object?

clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString

Can you instantiate the Math class?

You can't instantiate the math class. All the methods in this class are static. And the constructor is not public.

What is singleton?

It is one of the design pattern. This falls in the creational pattern of the design pattern. There will be only one instance for that entire JVM. You can achieve this by having the private constructor in the class. For e.g.., public class Singleton { private static final Singleton s = new Singleton(); private Singleton() { } public static Singleton getInstance() { return s; } // all non static methods … }

What is DriverManager?

The basic service to manage set of JDBC drivers.

What is Class.forName() does and how it is useful?

It loads the class into the ClassLoader. It returns the Class. Using that you can get the instance ("class-instance".newInstance()).

What is a Marker Interface?

An interface with no methods. Example: Serializable, Remote, And Cloneable

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 the purpose of abstract class?

It is not an instantiable class. It provides the concrete implementation for some/all the methods. So that they can reuse the concrete functionality by inheriting the abstract class.

What is the difference between interface and abstract class?

Abstract class defined with methods. Interface will declare only the methods. 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.

What do you mean by RMI and how it is useful?

RMI is a remote method invocation. Using RMI, you can work with remote object. The function calls are as though you are invoking a local variable. So it gives you a impression that you are working really with a object that resides within your own JVM though it is somewhere.

What is the protocol used by RMI?

RMI-IIOP

What is a hashCode?

hash code value for this object which is unique for every object.

What is a thread?

Thread is a block of code which can execute concurrently with other threads in the JVM.

What is the algorithm used in Thread scheduling?

Fixed priority scheduling.

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.

Is JDBC-ODBC bridge multi-threaded?

No

Does the JDBC-ODBC Bridge support multiple concurrent open statements per connection?

No

What is the use of serializable?

To persist the state of an object into any perminant storage device.

What is the use of transient?

It is an indicator to the JVM that those variables should not be persisted. It is the users responsibility to initialize the value when read back from the storage.

What are the different level lockings using the synchronization keyword?

Class level lock Object level lock Method level lock Block level lock

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.

Can a abstract method have the static qualifier?

No

What are the different types of qualifier and what is the default qualifier?

public, protected, private, package (default)

What is the super class of Hashtable?

Dictionary

What is a lightweight component?

Lightweight components are the one which doesn't go with the native call to obtain the graphical units. They share their parent component graphical units to render them. Example, Swing components

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 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.

How will you establish the connection between the servlet and an applet?

Using the URL, I will create the connection URL. Then by open Connection method of the URL, I will establish the connection, through which I can be able to exchange data.

What are the threads will start, when you start the java program?

Finalizer, Main, Reference Handler, Signal Dispatcher

How can you retrieve data from the ResultSet?

JDBC returns results in a ResultSet object, so we need to declare an instance of the class ResultSet to hold our results. The following code demonstrates declaring the ResultSet object rs.

ResultSet rs = stmt.executeQuery(”SELECT COF_NAME, PRICE FROM COFFEES”);
String s = rs.getString(”COF_NAME”);

The method getString is invoked on the ResultSet object rs, so getString() will retrieve (get) the value stored in the column COF_NAME in the current row of rs.

What are the different types of Statements?

Regular statement (use createStatement method), prepared statement (use preparedStatement method) and callable statement (use prepareCall)

How can you use PreparedStatement?

This special type of statement is derived from class Statement. If you need a Statement object to execute many times, it will normally make sense to use a PreparedStatement object instead. The advantage to this is that in most cases, this SQL statement will be sent to the DBMS right away, where it will be compiled. As a result, the PreparedStatement object contains not just an SQL statement, but an SQL statement that has been precompiled. This means that when the PreparedStatement is executed, the DBMS can just run the Preparedstatements SQL statement without having to compile it first.

PreparedStatement updateSales =

con.prepareStatement("UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ?");

What does setAutoCommit do?

When a connection is created, it is in auto-commit mode. This means that each individual SQL statement is treated as a transaction and will be automatically committed right after it is executed. The way to allow two or more statements to be grouped into a transaction is to disable auto-commit mode:

con.setAutoCommit(false);

Once auto-commit mode is disabled, no SQL statements will be committed until you call the method commit explicitly.

```
con.setAutoCommit(false);
PreparedStatement updateSales =
con.prepareStatement( "UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ?");
updateSales.setInt(1, 50); updateSales.setString(2, "Colombian");
updateSales.executeUpdate();
PreparedStatement updateTotal =
con.prepareStatement("UPDATE COFFEES SET TOTAL = TOTAL + ? WHERE COF_NAME LIKE ?");
updateTotal.setInt(1, 50);
updateTotal.setString(2, "Colombian");
updateTotal.executeUpdate();
con.commit();
con.setAutoCommit(true);
```

How do you call a stored procedure from JDBC?

The first step is to create a CallableStatement object. As with Statement an and PreparedStatement objects, this is done with an open Connection object. A CallableStatement object contains a call to a stored procedure.

```
CallableStatement cs = con.prepareCall("{call SHOW_SUPPLIERS}");
ResultSet rs = cs.executeQuery();
```

How do I retrieve warnings?

SQLWarning objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do; they simply alert the user that something did not happen as planned. A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method, which you must invoke in order to see the first warning reported on the calling object:

```
SQLWarning warning = stmt.getWarnings();
if (warning != null)
{
        System.out.println("n---Warning---n");
```

```
        while (warning != null)
        {
                System.out.println("Message: " + warning.getMessage());
                System.out.println("SQLState: " + warning.getSQLState());
                System.out.print("Vendor error code: ");
                System.out.println(warning.getErrorCode());
                System.out.println("");
                warning = warning.getNextWarning();
        }
}
```

How can you move the cursor in scrollable result sets?

One of the new features in the JDBC 2.0 API is the ability to move a result set's cursor backward as well as forward. There are also methods that let you move the cursor to a particular row and check the position of the cursor.

```
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
ResultSet srs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
```

The first argument is one of three constants added to the ResultSet API to indicate the type of a ResultSet object: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE , and TYPE_SCROLL_SENSITIVE. The second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable: CONCUR_READ_ONLY and CONCUR_UPDATABLE. The point to remember here is that if you specify a type, you must also specify whether it is read-only or updatable. Also, you must specify the type first, and because both parameters are of type int , the compiler will not complain if you switch the order. Specifying the constant TYPE_FORWARD_ONLY creates a nonscrollable result set, that is, one in which the cursor moves only forward. If you do not specify any constants for the type and updatability of a ResultSet object, you will automatically get one that is TYPE_FORWARD_ONLY and CONCUR_READ_ONLY.

What's the difference between TYPE_SCROLL_INSENSITIVE, and TYPE_SCROLL_SENSITIVE?

You will get a scrollable ResultSet object if you specify one of these ResultSet constants.The difference between the two has to do with whether a result set reflects changes that are made to it while it is open and whether certain methods can be called to detect these changes. Generally speaking, a result set that is

TYPE_SCROLL_INSENSITIVE does not reflect changes made while it is still open and one that is TYPE_SCROLL_SENSITIVE does. All three types of result sets will make changes visible if they are closed and then reopened:

```
Statement stmt =
        con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet srs =
        stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
srs.afterLast();
while (srs.previous())
{
        String name = srs.getString("COF_NAME");
        float price = srs.getFloat("PRICE");
        System.out.println(name + " " + price);
}
```

How to Make Updates to Updatable Result Sets?

Another new feature in the JDBC 2.0 API is the ability to update rows in a result set using methods in the Java programming language rather than having to send an SQL command. But before you can take advantage of this capability, you need to create a ResultSet object that is updatable. In order to do this, you supply the ResultSet constant CONCUR_UPDATABLE to the createStatement method.

```
Connection con =
        DriverManager.getConnection("jdbc:mySubprotocol:mySubName");
Statement stmt =
        con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
ResultSet uprs =
        stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
```

In Java, what is the difference between an Interface and an Abstract class?

An Abstract class declares have at least one instance method that is declared abstract which will be implemented by the subclasses. An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior.

Can you have virtual functions in Java? Yes or No. If yes, then what are virtual functions?

Yes, Java class functions are virtual by default. Virtual functions are functions of subclasses that can be invoked from a reference to their superclass. In other words, the functions of the actual object are called when a function is invoked on the reference to that object.

Write a function to reverse a linked list p in C++?

```
Link* reverse_list(Link* p){
if (p == NULL)
return NULL;
Link* h = p;
p = p->next;
h->next = NULL;
while (p != null){
Link* t = p->next;
p->next = h;
h = p;
p = t;
}
return h;
}
```

In C++, what is the usefulness of Virtual destructors?

Virtual destructors are neccessary to reclaim memory that were allocated for objects in the class hierarchy. If a pointer to a base class object is deleted, then the compiler guarantees the various subclass destructors are called in reverse order of the object construction chain.

What are mutex and semaphore? What is the difference between them?

A mutex is a synchronization object that allows only one process or thread to access a critical code block. A semaphore on the other hand allows one or more processes or threads to access a critial code block. A semaphore is a multiple mutex.

How many number of non-public class definitions can a source file have?

A source file can contain unlimited number of non-public class definitions

What types of values does boolean variables take?

It only takes values true and false

Which primitive data types are signed?

All except char and Boolean

Is char type signed or unsigned?

Char type is integral but unsigned. It range is 0 to 2^7-1

List primitive data types, there size and there range (min, max)

Data Type	Bytes	bits	min	max
boolean	-	1	-	-
char	2	16	0	2^16-1
byte	1	8	-2^7	2^7-1
short	2	16	-2^15	2^15-1
int	4	32	-2^31	2^31-1
long	8	64	-2^63	2^63-1
float	4	32	-	-
double	8	64	-	-

What forms an integral literal can be?

Decimal, octal and hexadecimal, hence example it can be 28, 034 and 0x1c respectively

What is the default value of Boolean?

False

Why is the main method static?

So that it can be invoked without creating an instance of that class

What is the difference between class variable, member variable and automatic (local) variable?

Class variable is a static variable and does not belong to instance of class but rather shared across all the instances

Member variable belongs to a particular instance of class and can be called from any method of the class

Automatic or local variable is created on entry to a method and has only method scope

When are static and non static variables of the class initialized?

The static variables are initialized when the class is loaded. Non static variables are initialized just before the constructor is called

When are automatic variable initialized?

Automatic variable have to be initialized explicitly

How is an argument passed in java, by copy or by reference?

If the variable is primitive datatype then it is passed by copy. If the variable is an object then it is passed by reference

What is garbage collection?

The runtime system keeps track of the memory that is allocated and is able to determine whether that memory is still useable. This work is usually done in background by a low-priority thread that is referred to as garbage collector. When the gc finds memory that is no longer accessible from any live thread it takes steps to release it back to the heap for reuse

Does System.gc and Runtime.gc() guarantee garbage collection?

No

What are different types of operators in java?

Uniary ++, --, +, -, |, ~, ()

Arithmetic *, /, %,+, -

Shift <<, >>, >>>

Comparison =, instanceof, = =,!=Bitwise &, ^, |Short Circuit &&, ||Ternary ?:Assignment =

How does bitwise (~) operator work?

It converts all the 1 bits in a binary value to 0s and all the 0 bits to 1s, e.g 11110000 coverts to 00001111

What is a modulo operator %?

This operator gives the value which is related to the remainder of a divisione.g x=7%4 gives remainder 3 as an answer

Can shift operators be applied to float types?

No, shift operators can be applied only to integer or long types

What happens to the bits that fall off after shifting?

They are discarded

What values of the bits are shifted in after the shift?

In case of signed left shift >> the new bits are set to zero. But in case of signed right shift it takes the value of most significant bit before the shift, that is if the most significant bit before shift is 0 it will introduce 0, else if it is 1, it will introduce 1

What are access modifiers?

These public, protected and private, these can be applied to class, variables, constructors and methods. But if you don’t specify an access modifier then it is considered as friendly.

Can protected or friendly features be accessed from different packages?

No when features are friendly or protected they can be accessed from all the classes in that package but not from classes in another package

How can you access protected features from another package?

You can access protected features from other classes by subclassing the that class in another package, but this cannot be done for friendly features

What are the rules for overriding?

Private method can be overridden by private, friendly, protected or public methods

Friendly method can be overridden by friendly, protected or public methods.

Protected method can be overridden by protected or public methods.

Public method can be overridden by public method

Explain modifier final.

Final can be applied to classes, methods and variables and the features cannot be changed. Final class cannot be subclassed, methods cannot be overridden

Can you change the reference of the final object?

No the reference cannot be change, but the data in that object can be changed

Can abstract modifier be applied to a variable?

No it is applied only to class and methods

Can abstract class be instantiated?

No abstract class cannot be instantiated i.e you cannot create a new object of this class

When does the compiler insist that the class must be abstract?

If one or more methods of the class are abstract, then the compiler will insist the class to be abstract. If class inherits one or more abstract methods from the parent abstract class and no implementation is provided for that method
If class implements an interface and provides no implementation for those methods

How is abstract class different from final class?

Abstract class must be subclassed and final class cannot be subclassed

Where can static modifiers be used?

They can be applied to variables, methods and even a block of code, static methods and variables are not associated with any instance of class

When are the static variables loaded into the memory?

During the class load time

When are the non static variables loaded into the memory?

They are loaded just before the constructor is called

Can static method use non static features of there class?

No they are not allowed to use non static features of the class, they can only call static methods and can use static data

What is static initializer code?

A class can have a block of initializer code that is simply surrounded by curly braces and labeled as static e.g.

```
public class Demo{
        static int =10;
        static{
                System.out.println("Hello                                world');
        }
}
```

And this code is executed exactly once at the time of class load

Where is native modifier used?

It can refer only to methods and it indicates that the body of the method is to be found else where and it is usually written in non java language

What are transient variables?

A transient variable is not stored as part of objects persistent state and they cannot be final or static

What is synchronized modifier used for?

It is used to control access of critical code in multithreaded programs

What are volatile variables?

It indicates that these variables can be modified asynchronously

What are wrapped classes?

Wrapped classes are classes that allow primitive types to be accessed as objects.

What are the four general cases for Conversion and Casting?

Conversion of primitives

Casting of primitives

Conversion of object references

Casting of object references

When can conversion happen?

It can happen during Assignment, Method call and Arithmetic promotion.

What are the rules for primitive assignment and method call conversion?

A boolean can not be converted to any other type

A non Boolean can be converted to another non boolean type, if the conversion is widening conversion

A non Boolean cannot be converted to another non boolean type, if the conversion is narrowing conversion

What are the rules for primitive arithmetic promotion conversion?

For Unary operators: If operant is byte, short or a char it is converted to an int and if it is any other type it is not converted

For binary operands: If one of the operands is double, the other operand is converted to double, Else If one of the operands is float, the other operand is converted to float, Else If one of the operands is long, the other operand is converted to long, Else both the operands are converted to int.

What are the rules for casting primitive types?

You can cast any non Boolean type to any other non boolean type
You cannot cast a boolean to any other type; you cannot cast any other type to a boolean.

What are the rules for object reference assignment and method call conversion?

An interface type can only be converted to an interface type or to object. If the new type is an interface, it must be a superinterface of the old type
A class type can be converted to a class type or to an interface type. If converting to a class type the new type should be superclass of the old type. If converting to an interface type new type the old class must implement the interface
An array maybe converted to class object, to the interface cloneable, or to an array. Only an array of object references types may be converted to an array, and the old element type must be convertible to the new element

What are the rules for Object reference casting?

Compile time rules:

When both Oldtypes and Newtypes are classes, one should be subclass of the other

When both Oldtype ad Newtype are arrays, both arrays must contain reference types (not primitive), and it must be legal to cast an element of Oldtype to an element of Newtype

You can always cast between an interface and a non-final object

Runtime rules:

If Newtype is a class. The class of the expression being converted must be Newtype or must inherit from Newtype

If NewType is an interface, the class of the expression being converted must implement Newtype.

What is the difference between while and do while loop?

Do while loop walways executes the body of the loop at least once, since the test is performed at the end of the body

When do you use continue and when do you use break statements?

When continue statement is applied it prematurely completes the iteration of a loop. When break statement is applied it causes the entire loop to be abandoned.

What is the base class from which all exceptions are subclasses?

All exceptions are subclasses of a class called java.lang.Throwable

How do you intercept and thereby control exceptions?

We can do this by using try/catch/finally blocks. You place the normal processing code in try block. You put the code to deal with exceptions that might arise in try block in catch block. Code that must be executed no matter what happens must be place in finally block

When do we say an exception is handled?

When an exception is thrown in a try block and is caught by a matching catch block, the exception is considered to have been handled

When do we say an exception is not handled?

There is no catch block that names either the class of exception that has been thrown or a class of exception that is a parent class of the one that has been thrown, then the exception is considered to be unhandled, in such condition the execution leaves the method directly as if no try has been executed

In what sequence does the finally block gets executed?

If you put finally after a try block without a matching catch block then it will be executed after the try block. If it is placed after the catch block and there is no exception then also it will be executed after the try block. If there is an exception and it is handled by the catch block then it will be executed after the catch block

What can prevent the execution of the code in finally block?

The death of thread

Use of system.exit()

Turning off the power to CPU

An exception arising in the finally block itself

What are the rules for catching multiple exceptions?

A more specific catch block must precede a more general one in the source, else it gives compilation error. Only one catch block, that is first applicable one, will be executed

What does throws statement declaration in a method indicate?

This indicates that the method throws some exception and the caller method should take care of handling it

What is checked exception?

Checked exceptions are exceptions that arise in a correct program, typically due to user mistakes like entering wrong data or I/O problems

What are runtime exceptions?

Runtime exceptions are due to programming bugs like out of bond arrays or null pointer exceptions.

What is difference between Exception and errors?

Errors are usually compile time and exceptions can be runtime or checked

How will you handle the checked exceptions?

You can provide a try/catch block to handle it. OR
Make sure method declaration includes a throws clause that informs the calling method an exception might be thrown from this particular method

When you extend a class and override a method, can this new method throw exceptions other than those that were declared by the original method?

No it cannot throw, except for the subclasses of those exceptions

Is it legal for the extending class which overrides a method which throws an exception, not o throw in the overridden class?

Yes it is perfectly legal

Explain the user defined Exceptions.

User defined Exceptions are the separate Exception classes defined by the user for specific purposed. An user defined can created by simply sub-classing it to the Exception class. This allows custom exceptions to be generated (using throw) and caught in the same way as normal exceptions.

Example:

```
class myCustomException extends Exception {
        // The class simply has to exist to be an exception
}
```

What's the difference between constructors and other methods?

Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.

What is the difference between Overloading and Overriding?

Overloading: Reusing the same method name with different arguments and perhaps a different return type is called as overloading

Overriding: Using the same method name with identical arguments and return type is know as overriding

What do you understand by late binding or virtual method Invocation (Example of runtime polymorphism)?

When a compiler for a non object oriented language comes across a method invocation, it determines exactly what target code should be called and build machine language to represent that call. In an object oriented language, this is not possible since the proper code to invoke is determined based upon the class if the object being used to make the call, not the type of the variable. Instead code is generated that will allow the decision to be made at run time. This delayed decision making is called as late binding

Can overriding methods have different return types?

No they cannot have different return types.

If the method to be overridden has access type protected, can subclass have the access type as private?

No, it must have access type as protected or public, since an overriding method must not be less accessible than the method it overrides

Can constructors be overloaded?

Yes constructors can be overloaded

What happens when a constructor of the subclass is called?

A constructor delays running its body until the parent parts of the class have been initialized. This commonly happens because of an implicit call to super() added by the compiler. You can provide your own call to super(arguments..) to control the way the parent parts are initialized. If you do this, it must be the first statement of the constructor.

If you use super() or this() in a constructor where should it appear in the constructor?

It should always be the first statement in the constructor

What is an inner class?

An inner class is same as any other class, but is declared inside some other class

How will you reference the inner class?

To reference it you will have to use OuterClass$InnerClass

Can objects that are instances of inner class access the members of the outer class?

Yes they can access the members of the outer class

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

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

Can inner classes be static?

Yes inner classes can be static, but they cannot access the non static data of the outer classes, though they can access the static data

Can an inner class be defined inside a method?

Yes it can be defined inside a method and it can access data of the enclosing methods or a formal parameter if it is final

What is an anonymous class?

Some classes defined inside a method do not need a name, such classes are called anonymous classes

What are the rules of anonymous class?

The class is instantiated and declared in the same place

The declaration and instantiation takes the form

new Xxxx () {// body} Where Xxxx is an interface name.

An anonymous class cannot have a constructor. Since you do not specify a name for the class, you cannot use that name to specify a constructor

Where does java thread support reside?

It resides in three places

The java.lang.Thread class (Most of the support resides here)

The java.lang.Object class

The java language and virtual machine

What is the difference between Thread and a Process?

Threads run inside process and they share data. One process can have multiple threads, if the process is killed all the threads inside it are killed, they dont share data

What happens when you call the start() method of the thread?

This registers the thread with a piece of system code called thread scheduler. The schedulers determines which thread is actually running

Does calling start () method of the thread causes it to run?

No it merely makes it eligible to run. The thread still has to wait for the CPU time along with the other threads, then at some time in future, the scheduler will permit the thread to run

When the thread gets to execute, what does it execute?

The thread executes a method call run(). It can execute run() method of either of the two choices given below:

The thread can execute it own run() method. The thread can execute the run() method of some other objects. For the first case you need to subclass the Thread class and give your subclass a run() method. For the second method you need to have a class implement the interface runnable. Define your run method. Pass this object as an argument to the Thread constructor

How many methods are declared in the interface runnable?

The runnable method declares only one method:

public void run();

Which way would you prefer to implement threading, by extending Thread class or implementing Runnable interface?

The preferred way will be to use Interface Runnable, because by subclassing the Thread class you have single inheritance i.e you wont be able to extend any other class

What happens when the run() method returns?

When the run() method returns, the thread has finished its task and is considered dead. You can't restart a dead thread. You can call the methods of dead thread

What are the different states of the thread?

They are as follows:

Running: The state that all thread aspire to be

Various waiting states: Waiting, Sleeping, Suspended and Blocked

Ready: Waiting only for the CPU

Dead: All done

What is Thread priority?

Every thread has a priority; the higher priorit thread gets preference over the lower priority thread by the thread scheduler

What is the range of priority integer?

It is from 1 to 10. 10 beings the highest priority and 1 being the lowest

What is the default priority of the thread?

The default priority is 5

What happens when you call Thread.yield()?

It caused the currently executing thread to move to the ready state if the scheduler is willing to run any other thread in place of the yielding thread. Yield is a static method of class Thread

What is the advantage of yielding?

It allows a time consuming thread to permit other threads to execute

What happens when you call Thread.sleep()?

It passes time without doing anything and without using the CPU. A call to sleep method requests the currently executing thread to cease executing for a specified amount of time.

Does the thread method start executing as soon as the sleep time is over?

No, after the specified time is over the thread enters into ready state and will only execute when the scheduler allows it to do so.

What do you mean by thread blocking?

If a method needs to wait an indeterminable amount of time until some I/O occurrence takes place, then a thread executing that method should graciously step out of the Running state. All java I/O methods behave this way. A thread that has graciously stepped out in this way is said to be blocked

What threading related methods are there in object class?

wait(), notify() and notifyAll() are all part of Object class and they have to be called from synchronized code only

What is preemptive scheduling?

In preemptive scheduling there are only two ways for the thread to leave the running state without ecplicitly calling wait() or suspended(). It can cease t be ready to execute ()by calling a blocking I/O method). It can get moved out by CPU by a higher priorit thread that becomes ready to execute

What is non-preemptive or Time sliced or round robin scheduling?

With time slicing the thread is allowd to execute for a limited amount of time. It is then moved to ready state, where it must contend with all the other ready threads.

What are the two ways of synchronizing the code?

Synchronizing an entire method by placing the synchronized modifier in the methods declaration. To execute the method, a thread must acquire the lock of the object that owns the method

Synchronize a subset of a method by surrounding the desired lines of code with curly brackets and inserting the synchronized expression before the opening curly. This allows you to synchronize the block on the lock of any object at all, not necessarily the object that owns the code

What happens when the wait() method is called?

The calling thread gives up CPU

The calling thread gives up the lock

The calling thread goes into the monitor's waiting pool

What is the ultimate ancestor of all java classes?

Object class is the ancestor of all the java classes

What are important methods of Object class?

wait(), notify(), notifyAll(), equals(), toString().

What is the difference between "= =" and "equals()"?

"= =" does shallow comparison, It retuns true if the two object points to the same address in the memory, i.e if the same the same reference "equals()" does deep comparison, it checks if the values of the data in the object are same

What would you use to compare two String variables - the operator == or the method equals()?

I'd use the method equals() to compare the values of the Strings and the == to check if two variables point at the same instance of a String object

Give example of a final class.

Math class is final class and hence cannot be extended

What is the difference between String and StringBuffer?

String is an immutable class, i.e you cannot change the values of that class. Example: String str = "java"; // address in memory say 12345

And now if you assign a new value to the variable str then str = "core java"; then the value of the variable at address 12345 will not change but a new memory is allocated for this variable say 54321, So in the memory

address 12345 will have value "java" and the memory address 54321 will have value "core java" and the variable str will now be pointing to address 54321 in memory.

StringBuffer can be modified dynamically. Example: StringBuffer strt ="java" // address in memory is say 12345 and now if you assign a new value to the variable str then Str = "core java"; then value in the address of memory will get replaced, a new memory address is not allocated in this case.

What will be the result if you compare StringBuffer with String if both have same values?

It will return false as you cannot compare String with StringBuffer

What is Collection API?

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 different types of collections?

A collection has no special order and does not reject duplicates

A list is ordered and does not reject duplicates

A set has no special order but rejects duplicates

A map supports searching on a key field, values of which must be unique

Tell something about Arrays

Arrays are fast to access, but are inefficient if the number of elements grows and if you have to insert or delete an element

What is the difference between ArrayList and Vector?

Vector methods are synchronized while ArrayList methods are not

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 the difference between Hashtable and HashMap?

Hashtable does not store null value, while HashMap does. Hashtable is synchronized, while HashMap is not

Why do we need public static void main(String args[]) method in Java?

We need

Public: The method can be accessed outside the class / package

Static: You need not have an instance of the class to access the method

Void: Your application need not return a value, as the JVM launcher would return the value when it exits

Main(): This is the entry point for the application

If the main() was not static, you would require an instance of the class in order to execute the method.

What is the difference between an Interface and an Abstract class?

In abstract class you can define as well as declare methods, the methods which are declared are to be marked as abstract. In interface all we just declare methods and the definition is provided by the class which is implementing it

Explain serialization.

Serialization means storing a state of a java object by coverting it to byte stream

What are the rules of serialization?

Static fileds are not serialized because they are not part of any one particular object

Fileds from the base class are handled only if hose are serializable

Transient fileds are not serialized

What is difference between error and exception?

Error occurs at runtime and cannot be recovered, Outofmemory is one such example. Exceptions on the other hand are due conditions which the application encounters such as FileNotFound exception or IO exceptions

What do you mean by object oriented programming?

In object oriented programming the emphasis is more on data than on the procedure and the program is divided into objects. The data fields are hidden and they cant be accessed by external functions. The design approach is bottom up. The functions operate on data that is tied together in data structure

What are 4 pillars of object oriented programming?

Abstraction: It means hiding the details and only exposing the essential parts

Polymorphism: Polymorphism means having many forms. In java you can see polymorphism when you have multiple methods with the same name

Inheritance: Inheritance means the child class inherits the non private properties of the parent class

Encapsulation: It means data hiding. In java with encapsulate the data by making it private and even we want some other class to work on that data then the setter and getter methods are provided

What is the difference between procedural and object oreinted language?

In procedural programming the instructions are executed one after another and the data is exposed to the whole program. In OOPs programming the unit of program is an object which is nothing but combination of data and code and the data is not exposed outside the object

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 is the difference between parameters and arguments?

While defining method, variables passed in the method are called parameters. While using those methods, values passed to those variables are called arguments.

What is reflection in java?

Reflection allows Java code to discover information about the fields, methods and constructors of loaded classes and to dynamically invoke them

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's the difference between a queue and a stack?

Stacks works by last-in-first-out rule (LIFO), while queues use the FIFO rule

Can you make an instance of abstract class?

No you cannot create an instance of abstract class

What are parsers?

Parsers are used for processing XML documents. There are 2 types of parsers DOM parser and SAX Parser

What is the difference between SAX and DOM parser?

DOM parsers are Object based and SAX parsers are event based DOM parsers creates Tree in the memory whereas SAX parser does not and hence it is faster than DOM. DOM parser are useful when we have to modify the XML, with SAX parser you cannot modify the xml, it is read only

What is the difference between Java Bean and Java Class?

Basically a Bean is a java class but it has getter and setter method and it does not have any logic in it, it is used for holding data. On the other hand the Java class can have what a java bean has and also has some logic inside it

What are null or Marker interfaces in Java?

The null interfaces are marker interfaces, they do not have function declarations in them, they are empty interfaces, this is to convey the compiler that they have to be treated differently

Does java Support multiple inheritance?

Java does not support multiple inheritance directly like C++, because then it is prone to ambiguity, example if a class extends 2 other classes and these 2 parent classes have same method names then there is ambiguity. Hence in Java Multiple inheritance is supported using Interfaces

What is virtual function?

In OOP when a derived class inherits from a base class, an object of the derived class may be referred to (or cast) as either being the base class type or the derived class type. If there are base class functions overridden by the derived class, a problem then arises when a derived object has been cast as the base class type. When a derived object is referred to as being of the base's type, the desired function call behavior is ambiguous. The distinction between virtual and not virtual is provided to solve this issue. If the function in question is designated "virtual" then the derived class's function would be called (if it exists). If it is not virtual, the base class's function would be called.

Does java support virtual functions?

No, java does not support virtual functions directly like in C++, but it supports using Abstract class and interfaces

Describe what happens when an object is created in Java.

Several things happen in a particular order to ensure the object is constructed properly:

Memory is allocated from heap to hold all instance variables and implementation-specific data of the object and its superclasses. Implemenation-specific data includes pointers to class and method data.

The instance variables of the objects are initialized to their default values.

The constructor for the most derived class is invoked. The first thing a constructor does is call the consctructor for its superclasses. This process continues until the constrcutor for java.lang.Object is called, as java.lang.Object is the base class for all objects in java.

Before the body of the constructor is executed, all instance variable initializers and initialization blocks are executed. Then the body of the constructor is executed.

Thus, the constructor for the base class completes first and constructor for the most derived class completes last.

What is the purpose of System Class?

The purpose of the system class is to provide the access to the System reources

What is instanceOf operator used for?

It is used to if an object can be cast into a specific type without throwing Class cast exception

Why we should not have instance variable in an interface?

Since all data fields and methods in an Interface are public by default, then we implement that interface in our class then we have public members in our class and this class will expose these data members and this is violation of encapsulation as now the data is not secure

What is JVM?

When we install a java package, it contains 2 things, Java Runtime Environment (JRE) Java Development Kit (JDK). The JRE provides runtime support for Java applications. The JDK provides the Java compiler and other development tools. The JDK includes the JRE. Both the JRE and the JDK include a Java Virtual Machine (JVM). This is the application that executes a Java program. A Java program requires a JVM to run on a particular platform.

Can abstract class be final?

No, abstract class cannot be final

When a new object of derived Class is created, whose constructor will be called first, childs or parents?

Even when the new object of child class is created, first the Base class constructor gets executed and then the child classes constructor

What is a singleton class?

A singleton is an object that cannot be instantiated. The restriction on the singleton is that there can be only one instance of a singleton created by the Java Virtual Machine (JVM) - by prevent direct instantiation we can ensure that developers don't create a second copy.

Can an abstract class have final method?

Yes

Can a final class have an abstract method?

No, the compiler will give an error

What is the difference between Authentication and Authorization?

Authentication is a process for verifying that an individual is who they say they are. Authorization is an additional level of security, and it means that a particular user (usually authenticated), may have access to a particular resource say record, file, directory or script.

What is JDBC?

Short for Java Database Connectivity, a Java API that enables Java programs to execute SQL statements. This allows Java programs to interact with any SQL-compliant database. Since nearly all relational database management systems (DBMSs) support SQL, and because Java itself runs on most platforms, JDBC makes it possible to write a single database application that can run on different platforms and interact with different DBMSs. JDBC is similar to ODBC, but is designed specifically for Java programs, whereas ODBC is language-independent.

What are the components of JDBC?

JDBC Components—Connection Pools, Data Sources, and MultiPools

How to you load the drivers?

Class.forName() method is used in JDBC to load the JDBC drivers dynamically

What does Class.forname() do?

Class.forName() method is used in JDBC to load the JDBC drivers dynamically

What are the different types of JDBC drivers?

A JDBC-ODBC bridge provides JDBC API access via one or more ODBC drivers. Note that some ODBC native code and in many cases native database client code must be loaded on each client machine that uses this type of driver. Hence, this kind of driver is generally most appropriate when automatic installation and downloading of a Java technology application is not important.

A native-API partly Java technology-enabled driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

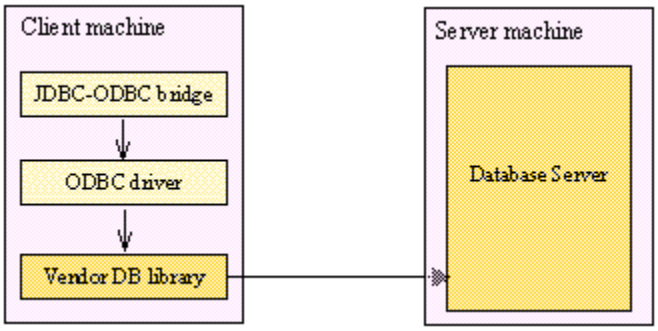
A net-protocol fully Java technology-enabled driver translates JDBC API calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect all of its Java technology-based clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC API alternative. It is likely that all vendors of this solution will provide products suitable for Intranet use. In order for

these products to also support Internet access they must handle the additional requirements for security, access through firewalls, etc., that the Web imposes. Several vendors are adding JDBC technology-based drivers to their existing database middleware products.

A native-protocol fully Java technology-enabled driver converts JDBC technology calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Intranet access. Since many of these protocols are proprietary the database vendors themselves will be the primary source for this style of driver. Several database vendors have these in progress.

What are the pros and cons of all the 4 drivers

Type 1: JDBC-ODBC Bridge: The type 1 driver, JDBC-ODBC Bridge, translates all JDBC calls into ODBC (Open DataBase Connectivity) calls and sends them to the ODBC driver. As such, the ODBC driver, as well as, in many cases, the client database code, must be present on the client machine. Figure 1 shows a typical JDBC-ODBC Bridge environment.

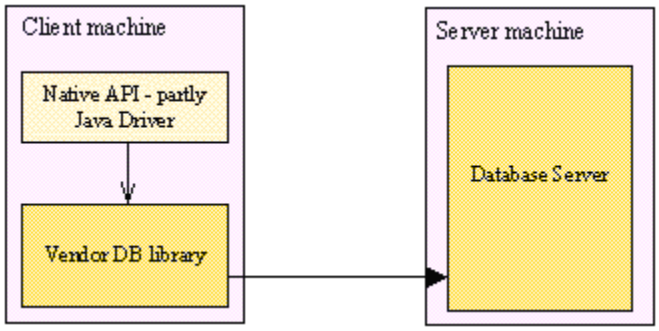


Pros: The JDBC-ODBC Bridge allows access to almost any database, since the database's ODBC drivers are already available. Type 1 drivers may be useful for those companies that have an ODBC driver already installed on client machines.

Cons: The performance is degraded since the JDBC call goes through the bridge to the ODBC driver, then to the native database connectivity interface. The result comes back through the reverse process. Considering the performance issue, type 1 drivers may not be suitable for large-scale applications. The ODBC driver and native connectivity interface must already be installed on the client machine. Thus any advantage of using Java applets in an intranet environment is lost, since the deployment problems of traditional applications remain.

Type 2: Native-API/partly Java driver: JDBC driver type 2 -- the native-API/partly Java driver -- converts JDBC calls into database-specific calls for

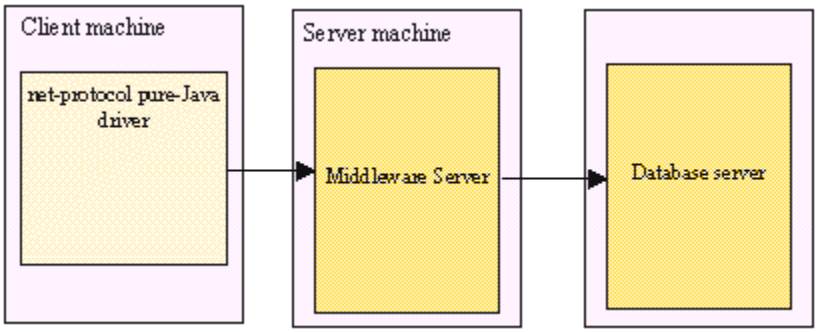
databases such as SQL Server, Informix, Oracle, or Sybase. The type 2 driver communicates directly with the database server; therefore it requires that some binary code be present on the client machine.



Pros: Type 2 drivers typically offer significantly better performance than the JDBC-ODBC Bridge.

Cons: The vendor database library needs to be loaded on each client machine. Consequently, type 2 drivers cannot be used for the Internet. Type 2 drivers show lower performance than type 3 and type 4 drivers.

Type 3: Net-protocol/all-Java driver: JDBC driver type 3 -- the net-protocol/all-Java driver -- follows a three-tiered approach whereby the JDBC database requests are passed through the network to the middle-tier server. The middle-tier server then translates the request (directly or indirectly) to the database-specific native-connectivity interface to further the request to the database server. If the middle-tier server is written in Java, it can use a type 1 or type 2 JDBC driver to do this.

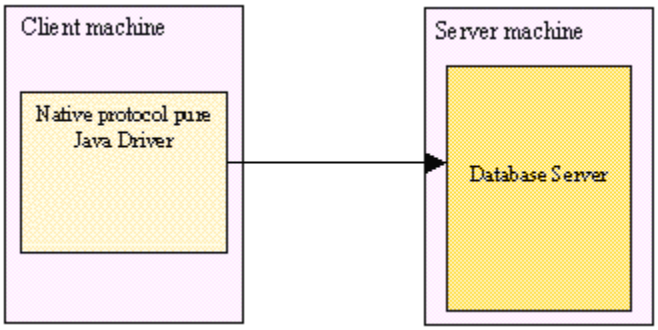


Pros: The net-protocol/all-Java driver is server-based, so there is no need for any vendor database library to be present on client machines. Further, there are many opportunities to optimize portability, performance, and scalability. Moreover, the net protocol can be designed to make the client JDBC driver very small and fast to load. Additionally, a type 3 driver typically provides support for features such as

caching (connections, query results, and so on), load balancing, and advanced system administration such as logging and auditing.

Cons: Type 3 drivers require database-specific coding to be done in the middle tier. Additionally, traversing the recordset may take longer, since the data comes through the backend server.

Type 4: Native-protocol/all-Java driver: The native-protocol/all-Java driver (JDBC driver type 4) converts JDBC calls into the vendor-specific database management system (DBMS) protocol so that client applications can communicate directly with the database server. Level 4 drivers are completely implemented in Java to achieve platform independence and eliminate deployment administration issues.



Pros: Since type 4 JDBC drivers don't have to translate database requests to ODBC or a native connectivity interface or to pass the request on to another server, performance is typically quite good. Moreover, the native-protocol/all-Java driver boasts better performance than types 1 and 2. Also, there's no need to install special software on the client or server. Further, these drivers can be downloaded dynamically.

Cons: With type 4 drivers, the user needs a different driver for each database.

How to you establish a connection?

Loading Drivers

Class.forName("Driver");

Getting connection

Connection con = DriverManager.getConnection(url, ”myLogin", "myPassword");

What are different types of statements in JDBC?

java.sql.Statement: Top most interface which provides basic methods useful for executing SELECT, INSERT, UPDATE and DELETE SQL statements.

java.sql.PreparedStatement: An enhanced verion of java.sql.Statement which allows precompiled queries with parameters. It is more efficient to use java.sql.PreparedStatement if you have to specify parameters to your SQL queries.

java.sql.CallableStatement: Allows you to execute stored procedures within a RDBMS which supports stored procedures (MySQL doesn't support stored procedures at the moment).

When do we use prepared statements?

If you want to execute a Statement object many times, it will normally reduce execution time to use a PreparedStatement object instead. The main feature of a PreparedStatement object is that, unlike a Statement object, it is given an SQL statement when it is created. The advantage to this is that in most cases, this SQL statement will be sent to the DBMS right away, where it will be compiled. As a result, the PreparedStatement object contains not just an SQL statement, but an SQL statement that has been precompiled. This means that when the PreparedStatement is executed, the DBMS can just run the PreparedStatement 's SQL statement without having to compile it first. Although PreparedStatement objects can be used for SQL statements with no parameters, you will probably use them most often for SQL statements that take parameters. The advantage of using SQL statements that take parameters is that you can use the same statement and supply it with different values each time you execute it.

How do you create JDBC statements?

```
Connection con = null;
Statement st = null;
// Obtain connection here
st = con.createStatement();
ResultSet rs = null;
rs = st.executeQuery("SELECT * FROM users");
int recordsUpdated;
recordsUpdated = st.executeUpdate("DELETE FROM users WHERE user_id = 1");
```

How do you retrieve data from a result set?

Example:

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(SELECT COF_NAME, PRICE FROM COFFEES”);
```

```
while (rs .next() ) {
        //Iam assuming there are 3 columns in the table.
        System.out.println ( rs.getString(1));
        System.out.println(rs.getString(2));
        System.out.println(rs.getString(3));
}
//don't forget to close the resultset, statement & connection
rs.close(); //First
stmt.close(); //Second
con.close(); //Last
System.out.println("You are done");
```

What is a stored procedure?

A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. Stored procedures are used to encapsulate a set of operations or queries to execute on a database server. For example, operations on an employee database (hire, fire, promote, lookup) could be coded as stored procedures executed by application code. Stored procedures can be compiled and executed with different parameters and results, and they may have any combination of input, output, and input/output parameters.

What are the tasks of JDBC?

Load the JDBC drivers

Register the drivers

Specify a database

Open a connection to database

Submit a query to database

Gets the results

What do you mean by batch updates?

If you want to execute a set of statements, i.e. SQL statements at a time then we use batch update statement.

resultset=pst.batchUpdate();

Why do we need batch updates?

Let's say there are 100 records need to be insert. If we execute normal statemets the no of transactions will be 100 (in terms of connection making to DB). Using batch updates we can add 100 rec to batch and the no of transactions will be

only one in this case. This will reduce the burdon on db, which is very costly in terms of resources.

How do you call a stored procedure from java?

You can call a stored procedure using Callable statements.

```
CallableStatement cs = con.prepareCall("{call StoredProc}");
ResultSet rs = cs.executeQuery();
```

What packages are being used by JDBC?

Following packages are used in JDBC

java.sql, javax.sql

Explain how to get the resultset of Stored procedure?

```
CallableStatement cstmt;
ResultSet rs;
int i;
String s;
...
cstmt.execute();// Call the stored procedure  1
rs = cstmt.getResultSet();// Get the first result set  2
while (rs.next()) {              // Position the cursor  3
      i = rs.getInt(1);          // Retrieve current result set value
      System.out.println("Value from first result set = " + i);   // Print the value
}
cstmt.getMoreResults();  // Point to the second result set  4a
                                  // and close the first result set
rs = cstmt.getResultSet(); // Get the second result set  4b
while (rs.next()) {              // Position the cursor  4c
      s = rs.getString(1);   // Retrieve current result set value
      System.out.println("Value from second result set = " + s);
                                  // Print the value
}
rs.close();                      // Close the result set
cstmt.close();                  // Close the statement
```

What do we use setAutoCommit() for?

The DML operations by default are committed. If we wish to avoid the commit by default, setAutoCommit(false) has to be called on the Connection object. Once the

statements are executed, commit() has to be called on the Connection object explicitly.

What is the difference between Resultset and Rowset?

RowSet: The interface that adds support to the JDBC API for the javaBeansTM component model. A rowset, which can be used as a JavaBeans component in a visual Bean development environment, can be created and configured at design time and executed at run time. The RowSet interface provides a set of JavaBeans properties that allow a RowSet instance to be configured to connect to a JDBC data source and read some data from the data source. A group of setter methods (setInt, setBytes, setString, and so on) provide a way to pass input parameters to a rowset's command property. This command is the SQL query the rowset uses when it gets its data from a relational database, which is generally the case. The RowSet interface supports JavaBeans events, allowing other components in an application to be notified when an event occurs on a rowset, such as a change in its value. The RowSet interface is unique in that it is intended to be implemented using the rest of the JDBC API. In other words, a RowSet implementation is a layer of software that executes "on top" of a JDBC driver. Implementations of the RowSet interface can be provided by anyone, including JDBC driver vendors who want to provide a RowSet implementation as part of their JDBC products. A RowSet object may make a connection with a data source and maintain that connection throughout its life cycle, in which case it is called a connected rowset. A rowset may also make a connection with a data source, get data from it, and then close the connection. Such a rowset is called a disconnected rowset. A disconnected rowset may make changes to its data while it is disconnected and then send the changes back to the original source of the data, but it must reestablish a connection to do so.

ResultSet: A table of data representing a database result set, which is usually generated by executing a statement that queries the database. A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set. A default ResultSet object is not updatable and has a cursor that moves forward only. Thus, you can iterate through it only once and only from the first row to the last row. It is possible to produce ResultSet objects that are scrollable and/or updatable. The following code fragment, in which con is a valid Connection object, illustrates how to make a result

set that is scrollable and insensitive to updates by others, and that is updatable. See ResultSet fields for other options.

How do we retrieve warning?

SQLWarning objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do. They simply alert the user that something did not happen as planned. A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method, which you must invoke in order to see the first warning reported on the calling object.

Example:

```
SQLWarning warning = stmt.getWarnings();
if (warning != null) {
        while (warning != null) {
                System.out.println("Message: " + warning.getMessage());
                System.out.println("SQLState: " + warning.getSQLState());
                System.out.print("Vendor error code: ");
                System.out.println(warning.getErrorCode());
                warning = warning.getNextWarning();
        }
}
```

How many statements can we create with one connection?

There is no such limit on number of statements to be created

How to Make Updates to Updatable Result Sets?

Using JDBC 2.0 API we have the ability to update rows in a result set. For this we need to create a ResultSet object that is updatable. For this, we pass the ResultSet constant CONCUR_UPDATABLE to the createStatement method.

```
Connection con = DriverManager.getConnection(url, "myLogin", "myPassword");
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
ResultSet uprs = stmt.executeQuery("SELECT NAME, SALARY FROM EMPLOYEES");
```

How can you move the cursor in scrollable result sets?

In JDBC 2.0 API we have the ability to move a result set's cursor backward as well as forward. We can also move the cursor to a particular row and check the position of the cursor.

Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);

ResultSet srs = stmt.executeQuery("SELECT NAME, SALARY FROM EMPLOYEES");

The first argument is one of three constants added to the ResultSet API to indicate the type of a ResultSet object: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE , and TYPE_SCROLL_SENSITIVE. The second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable: CONCUR_READ_ONLY and CONCUR_UPDATABLE. Make sure that when you specify a type, you must also specify whether it is read-only or updatable. Specifying the constant TYPE_FORWARD_ONLY creates a nonscrollable result set, that is, one in which the cursor moves only forward. If you do not specify any constants for the type and updatability of a ResultSet object, you will automatically get one that is TYPE_FORWARD_ONLY and CONCUR_READ_ONLY.

What's the difference between TYPE_SCROLL_INSENSITIVE and TYPE_SCROLL_SENSITIVE?

You will get a scrollable ResultSet object if you specify one of these ResultSet constants.The difference between the two has to do with whether a result set reflects changes that are made to it while it is open and whether certain methods can be called to detect these changes. Generally speaking, a result set that is TYPE_SCROLL_INSENSITIVE does not reflect changes made while it is still open and one that is TYPE_SCROLL_SENSITIVE does. All three types of result sets will make changes visible if they are closed and then reopened:

```
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet srs = stmt.executeQuery("SELECT NAME, SALARY FROM PERSON");
srs.afterLast();
while (srs.previous()){
    String name = srs.getString("NAME");
    float salary = srs.getFloat("SALARY");
    System.out.println(name + " " + salary);}
```

How do you insert images in Database using JDBC?

We can store images in the database using the BLOB datatype where in the image is stored as a byte stream

What is Metadata?

It is information about one of two things: Database information (java.sql.DatabaseMetaData), or Information about a specific ResultSet (java.sql.ResultSetMetaData). Use DatabaseMetaData to find information about your database, such as its capabilities and structure. Use ResultSetMetaData to find information about the results of an SQL query, such as size and types of columns

What is a data source?

A DataSource class brings another level of abstraction than directly using a connection object. Data source can be referenced by JNDI. Data Source may point to RDBMS, file System , any DBMS etc.