Java Interview Questions and Answers

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.

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

What is similarities/difference between an Abstract class and Interface?
Differences are as follows:
Interfaces provide a form of multiple inheritance. A class can extend only one other class. Interfaces are limited to public methods and constants with no implementation. Abstract classes can have a partial implementation, protected parts, static methods, etc.
A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class. Interfaces are slow as it requires extra indirection to to find corresponding method in in the actual class. Abstract classes are fast.
Similarities:

Neither Abstract classes or Interface can be instantiated.

Java Interview Questions - How to define an Abstract class?
A class containing abstract method is called Abstract class. An Abstract class can't be instantiated.
Example of Abstract class:

```
abstract class testAbstractClass {
protected String myString;
public String getMyString() {
return myString;
}
public abstract string anyAbstractFunction();
}
```

How to define an Interface in Java ?
In Java Interface defines the methods but does not implement them. Interface can include constants. A class that implements the interfaces is bound to implement all the methods defined in Interface.
Emaple of Interface:

```
public interface sampleInterface {
public void functionOne();

public long CONSTANT_ONE = 1000;
}
```

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

How many methods in the Serializable interface?
There is no method in the Serializable interface. The Serializable interface acts as a marker, telling the object serialization tools that your class is serializable.

How many methods in the Externalizable interface?
There are two methods in the Externalizable interface. You have to implement these two methods in order to make your class externalizable. These two methods are readExternal() and writeExternal().

What is the difference between Serializalble and Externalizable interface?
When you use Serializable interface, your class is serialized automatically by default. But you can override writeObject() and readObject() two methods to control more complex object serailization process. When you use Externalizable interface, you have a complete control over your class's serialization process.

What is a transient variable in Java?
A transient variable is a variable that may not be serialized. If you don't want some field to be serialized, you can mark that field transient or static.

Which containers use a border layout as their default layout?
The Window, Frame and Dialog classes use a border layout as their default layout.

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 Java?
Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C. Java should not be confused with JavaScript, which shares only the name and a similar C-like syntax. Sun Microsystems currently maintains and updates Java regularly.

What does a well-written OO program look like?
A well-written OO program exhibits recurring structures that promote abstraction, flexibility, modularity and elegance.

Can you have virtual functions in Java?
Yes, all functions in Java are virtual by default. This is actually a pseudo trick question because the word "virtual" is not part of the naming convention in Java (as it is in C++, C-sharp and VB.NET), so this would be a foreign concept for someone who has only coded in Java. Virtual functions or virtual methods are functions or methods that will be redefined in derived classes.

Jack developed a program by using a Map container to hold key/value pairs. He wanted to make a change to the map. He decided to make a clone of the map in order to save the original data on side. What do you think of it? ?
If Jack made a clone of the map, any changes to the clone or the original map would be seen on both maps, because the clone of Map is a shallow copy. So Jack made a wrong decision.

What is more advisable to create a thread, by implementing a Runnable interface or by extending Thread class?
Strategically speaking, threads created by implementing Runnable interface are more advisable. If you create a thread by extending a thread class, you cannot extend any other class. If you create a thread by implementing Runnable interface, you save a space for your class to extend another class now or in future.

What is NullPointerException and how to handle it?

When an object is not initialized, the default value is null. When the following things happen, the NullPointerException is thrown:
--Calling the instance method of a null object.
--Accessing or modifying the field of a null object.
--Taking the length of a null as if it were an array.
--Accessing or modifying the slots of null as if it were an array.
--Throwing null as if it were a Throwable value.
The NullPointerException is a runtime exception. The best practice is to catch such exception even if it is not required by language design.

An application needs to load a library before it starts to run, how to code?
One option is to use a static block to load a library before anything is called. For example,

```
class Test {
static {
System.loadLibrary("path-to-library-file");
}
....
}
```

When you call new Test(), the static block will be called first before any initialization happens. Note that the static block position may matter.

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("output.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.

Name the containers which uses Border Layout as their default layout?
Containers which uses Border Layout as their default are: window, Frame and Dialog classes.

What do you understand by Synchronization?
Synchronization is a process of controlling the access of shared resources by the multiple threads in such a manner that only one thread can access one resource at a time. In non synchronized multithreaded application, it is possible for one thread to modify a shared object while another thread is in the process of using or updating the object's value.
Synchronization prevents such type of data corruption.
E.g. Synchronizing a function:

```
public synchronized void Method1 () {
// Appropriate method-related code.
}
```

E.g. Synchronizing a block of code inside a function:

```
public myFunction (){
synchronized (this) {
// Synchronized code here.
}
}
```

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 causes dirty data and leads to significant errors.

What are synchronized methods and synchronized statements?
Synchronized methods are methods that are used to control access to a method or 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 three ways in which a thread can enter the waiting state?
A thread can enter the waiting state by invoking its sleep() method, by blocking on IO, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

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's new with the stop(), suspend() and resume() methods in JDK 1.2?
The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

What is the preferred size of a component?
The preferred size of a component is the minimum component size that will allow the component to display normally.

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.

What is thread?
A thread is an independent path of execution in a system.

What is multi-threading?
Multi-threading means various threads that run in a system.

How does multi-threading take place on a computer with a single CPU?
The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

How to create a thread in a program?
You have two ways to do so. First, making your class "extends" Thread class. Second, making your class "implements" Runnable interface. Put jobs in a run() method and call start() method to start the thread.

Can Java object be locked down for exclusive use by a given thread?
Yes. You can lock an object by putting it in a "synchronized" block. The locked object is inaccessible to any thread other than the one that explicitly claimed it.

Can each Java object keep track of all the threads that want to exclusively access to it?
Yes. Use Thread.currentThread() method to track the accessing thread.

Does it matter in what order catch statements for FileNotFoundException and IOExceptipon are written?
Yes, it does. The FileNoFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

What invokes a thread's run() method?
After a thread is started, via its start() method of the Thread class, the JVM invokes the thread's run() method when the thread is initially executed.

What is the purpose of the wait(), notify(), and notifyAll() methods?
The wait(),notify(), and notifyAll() methods are used to provide an efficient way for threads to communicate each other.

What are the high-level thread states?
The high-level thread states are ready, running, waiting, and dead.

What is the difference between yielding and sleeping?
When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

What happens when a thread cannot acquire a lock on an object?
If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

What is the difference between Process and Thread?
A process can contain multiple threads. In most multithreading operating systems, a process gets its own memory address space; a thread doesn't. Threads typically share the heap belonging to their parent process. For instance, a JVM runs in a single process in the host O/S. Threads in the JVM share the heap belonging to that process; that's why several threads may access the same object. Typically, even though they share a common heap, threads have their own stack space. This is how one thread's invocation of a method is kept separate from another's. This is all a gross oversimplification, but it's accurate enough at a high level. Lots of details differ between operating systems. Process vs. Thread A program vs. similar to a sequential program an run on its own vs. Cannot run on its own Unit of allocation vs. Unit of execution Have its own memory space vs. Share with others Each process has one or more threads vs. Each thread belongs to one process Expensive, need to context switch vs. Cheap, can use process memory and may not need to context switch More secure. One process cannot corrupt another process vs. Less secure. A thread can write the memory used by another thread

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 &emp;&emp; with &emp; in the following code: String a=null; if (a!=null && a.length()>10) {...}
A single ampersand here would lead to a NullPointerException.

What is the Vector class?
The Vector class provides the capability to implement a growable array of objects

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.

If a method is declared as protected, where may the method be 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.

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

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's the main difference between a Vector and an ArrayList?
Java Vector class is internally synchronized and ArrayList is not.

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

Does garbage collection guarantee that a program will not run out of memory?
No, it doesn't. 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.

Name Component subclasses that support painting ?
The Canvas, Frame, Panel, and Applet classes support painting.

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

How can you write a loop indefinitely?
for(;;)--for loop; while(true)--always true, etc.

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 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.

When should the method invokeLater()be used?
This method is used to ensure that Swing components are updated through the event-dispatching thread.

How many methods in Object class?
This question is not asked to test your memory. It tests you how well you know Java. Ten in total.
clone()
equals() & hashcode()
getClass()
finalize()
wait() & notify()
toString()

How does Java handle integer overflows and underflows?
It uses low order bytes of the result that can fit into the size of the type allowed by the operation.

What is the numeric promotion?
Numeric promotion is used with both unary and binary bitwise operators. This means that byte, char, and short values are converted to int values before a bitwise operator is applied.
If a binary bitwise operator has one long operand, the other operand is converted to a long value.
The type of the result of a bitwise operation is the type to which the operands have been promoted. For example:
short a = 5;
byte b = 10;
long c = 15;
The type of the result of (a+b) is int, not short or byte. The type of the result of (a+c) or (b+c) is long.

Is the numeric promotion available in other platform?
Yes. Because Java is implemented using a platform-independent virtual machine, bitwise operations always yield the same result, even when run on machines that use radically different CPUs.

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.
Operator & has no chance to skip both sides evaluation and && operator does. If asked why, give details as above.

When is the ArithmeticException throwQuestion: What is the GregorianCalendar class?
The GregorianCalendar provides support for traditional Western calendars.

What is the SimpleTimeZone class?
The SimpleTimeZone class provides support for a Gregorian calendar.

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 is the Properties class?
The properties class is a subclass of Hashtable that can be read from or written to a stream. It also provides the capability to specify a set of default values to be used.

What is the purpose of the Runtime class?
The purpose of the Runtime class is to provide access to the Java runtime system.

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

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

What is an abstract method?
An abstract method is a method whose implementation is deferred to a subclass. Or, a method that has no implementation.

What is the difference between interface and abstract class?
interface contains methods that must be abstract; abstract class may contain concrete methods. interface contains variables that must be static and final; abstract class may contain non-final and final variables. members in an interface are public by default, abstract class may contain non-public members. interface is used to "implements"; whereas abstract class is used to "extends". interface can be used to achieve multiple inheritance; abstract class can be used as a single inheritance. interface can "extends" another interface, abstract class can "extends" another class and "implements" multiple interfaces. interface is absolutely abstract; abstract class can be invoked if a main() exists. interface is more flexible than abstract class because one class can only "extends" one super class, but "implements" multiple interfaces. If given a choice, use interface instead of abstract class.

What is a static method?
A static method is a method that belongs to the class rather than any object of the class and doesn't apply to an object or even require that any objects of the class have been instantiated.

What is a protected method?
A protected method is a method that can be accessed by any method in its package and inherited by any subclass of its class.

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.

What is an object's lock and which object's 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.

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

What is the difference between a Window and a Frame?
The Frame class extends Window to define a main application window that can have a menu bar.

What is the difference between a Window and a Frame?
Heavy weight components like Abstract Window Toolkit (AWT), depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button. In this relationship, the Motif button is called the peer to the java.awt.Button. If you create two Buttons, two peers and hence two Motif Buttons are also created. The Java platform communicates with the Motif Buttons using the Java Native Interface. For each and every component added to the application, there is an additional overhead tied to the local windowing system, which is why these components are called heavy weight.

Which package has light weight components?
javax.Swing package. All components in Swing, except JApplet, JDialog, JFrame and JWindow are lightweight components.

What are peerless components?
The peerless components are called light weight components.

What is the difference between the Font and FontMetrics classes?
The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object

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.

What is the difference between throw and throws keywords?
The throw keyword denotes a statement that causes an exception to be initiated. It takes the Exception object to be thrown as argument. The exception will be caught by an immediately encompassing try-catch construction or propagated further up the calling hierarchy.
The throws keyword is a modifier of a method that designates that exceptions may come out of the method, either by virtue of the method throwing the exception itself or because it fails to catch such exceptions that a method it calls may throw.

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 or friendly access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

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

Does a class inherit the constructors of its super class?
A class does not inherit constructors from any of its superclasses.

Name primitive Java types.
The 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.

How can a GUI component handle its own events?
A component can handle its own events by implementing the required event-listener interface and adding itself as its own event listener.

How are the elements of a GridBagLayout organized?
The elements of a GridBagLayout are organized according to a grid. However, the elements are of different sizes and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

What advantage do Java's layout managers provide over traditional windowing systems?
Java uses layout managers to lay out components in a consistent manner across all windowing platforms. Since Java's layout managers aren't tied to absolute sizing and positioning, they are able to accommodate platform-specific differences among windowing systems.

What are the problems faced by Java programmers who don't use layout managers?
Without layout managers, Java programmers are faced with determining how their GUI will be displayed across multiple windowing systems and finding a common sizing and positioning that will work within the constraints imposed by each windowing system.

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.

What is the difference between the paint() and repaint() methods?
The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.

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.

Why would you use a synchronized block vs. synchronized method?
Synchronized blocks place locks for shorter periods than synchronized methods.

What restrictions are placed on method overriding?
Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden method.

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.

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).

What class allows you to read objects directly from a stream?
The ObjectInputStream class supports the reading of objects from input streams.

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.

How is it possible for two String objects with identical values not to be equal under the == operator?
How are this() and super() used with constructors?
The == operator compares two objects to determine if they are the same objects in memory. It is possible for two String objects to have the same value, but located in different areas of memory.

What is an IO filter?
An IO 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 the Set interface?
The Set interface provides methods for accessing the elements of a finite mathematical set. Sets do not allow duplicate elements.

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.

What is the purpose of the enableEvents() method?
The enableEvents() method is used to enable an event for a particular object. Normally, an event is enabled when a listener is added to an object for a particular event. The enableEvents() method is used by objects that handle events by overriding their event-dispatch methods.

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 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.

What is the ResourceBundle class?
The ResourceBundle class is used to store locale-specific resources that can be loaded by a program to tailor the program's appearance to the particular locale in which it is being run.

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 supeclass type, the casting is performed automatically.

What is a Java package and how is it used?
A Java package is a naming context for classes and interfaces. A package is used to create a separate name space for groups of classes and interfaces. Packages are also used to organize related classes and interfaces into a single API unit and to control accessibility to these classes and interfaces.

How do you restrict a user to cut and paste from the html page?
Using Servlet or client side scripts to lock keyboard keys. It is one of solutions.

What are the Object and Class classes used for?
The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program.

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.

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.

Does the code in finally block get executed if there is an exception and a return statement in a catch block?
If an exception occurs and there is a return statement in catch block, the finally block is still executed. The finally block will not be executed when the System.exit(1) statement is executed earlier or the system shut down earlier or the memory is used up earlier before the thread goes to finally block.

Is Java a super set of JavaScript?
No. They are completely different. Some syntax may be similar.

What is a Container in a GUI?
A Container contains and arranges other components (including other containers) through the use of layout managers, which use specific layout policies to determine where components should go as a function of the size of the container.

How the object oriented approach helps us keep complexity of software development under control?
We can discuss such issue from the following aspects:
Objects allow procedures to be encapsulated with their data to reduce potential interference.
Inheritance allows well-tested procedures to be reused and enables changes to make once and have effect in all relevant places.
The well-defined separations of interface and implementation allow constraints to be imposed on inheriting classes while still allowing the flexibility of overriding and overloading.

What is polymorphism?
Polymorphism means "having many forms". It allows methods (may be variables) to be written that needn't be concerned about the specifics of the objects they will be applied to. That is, the method can be specified at a higher level of abstraction and can be counted on to work even on objects of un-conceived classes.

What is design by contract?
The design by contract specifies the obligations of a method to any other methods that may use its services and also theirs to it. For example, the preconditions specify what the method required to be true when the method is called. Hence making sure that preconditions are. Similarly, postconditions specify what must be true when the method is finished, thus the called method has the responsibility of satisfying the post conditions.
In Java, the exception handling facilities support the use of design by contract, especially in the case of checked exceptions. The assert keyword can be used to make such contracts.

What are use cases?
A use case describes a situation that a program might encounter and what behavior the program should exhibit in that circumstance. It is part of the analysis of a program. The collection of use cases should, ideally, anticipate all the standard circumstances and many of the extraordinary circumstances possible so that the program will be robust.

What is scalability and performance?
Performance is a measure of "how fast can you perform this task." and scalability describes how an application behaves as its workload and available computing resources increase.

What is the benefit of subclass?
Generally: The sub class inherits all the public methods and the implementation.
The sub class inherits all the protected methods and their implementation.
The sub class inherits all the default(non-access modifier) methods and their implementation.
The sub class also inherits all the public, protected and default member variables from the super class.
The constructors are not part of this inheritance model.

How to add menushortcut to menu item?
If you have a button instance called aboutButton, you may add menu short cut by calling aboutButton.setMnemonic('A'), so the user may be able to use Alt+A to click the button.

In System.out.println(),what is System,out and println,pls explain?
System is a predefined final class,out is a PrintStream object acting as a field member and println is a built-in overloaded method in the out object.

Can you write a Java class that could be used both as an applet as well as an application?
A. Yes. Add a main() method to the applet.

Can you make an instance of an abstract class? For example - java.util.Calender is an abstract class with a method getInstance() which returns an instance of the Calender class.
No! You cannot make an instance of an abstract class. An abstract class has to be sub-classed. If you have an abstract class and you want to use a method which has been implemented, you may need to subclass that abstract class, instantiate your subclass and then call that method.

What is the output of x > y? a:b = p*q when x=1,y=2,p=3,q=4?
When this kind of question has been asked, find the problems you think is necessary to ask back before you give an answer. Ask if variables a and b have been declared or initialized. If the answer is yes. You can say that the syntax is wrong. If the statement is rewritten as: x

What is the difference between Swing and AWT components?
AWT components are heavy-weight, whereas Swing components are lightweight. Heavy weight components depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button.

Why Java does not support pointers?
Because pointers are unsafe. Java uses reference types to hide pointers and programmers feel easier to deal with reference types without pointers. This is why Java and C-sharp shine.

Parsers? DOM vs SAX parser
Parsers are fundamental xml components, a bridge between XML documents and applications that process that XML. The parser is responsible for handling xml syntax, checking the contents of the document against constraints established in a DTD or Schema.
DOM
1. Tree of nodes
2. Memory: Occupies more memory, preffered for small XML documents
3. Slower at runtime
4. Stored as objects
5. Programmatically easy
6. Ease of navigation
SAX
1. Sequence of events
2. Doesn't use any memory preferred for large documents
3. Faster at runtime
4. Objects are to be created
5. Need to write code for creating objects
6. Backward navigation is not possible as it sequentially processes the document

Can you declare a class as private?
Yes, we can declare a private class as an inner class. For example,

```
class MyPrivate {
private static class MyKey {
String key = "12345";
}
public static void main(String[] args) {
System.out.println(new MyKey().key);//prints 12345
}
}
```

What is the difference between shallow copy and deep copy?
Shallow copy shares the same reference with the original object like cloning, whereas the deep copy get a duplicate instance of the original object. If the shallow copy has been changed, the original object will be reflected and vice versa.

Can one create a method which gets a String and modifies it?
No. In Java, Strings are constant or immutable; their values cannot be changed after they are created, but they can be shared. Once you change a string, you actually create a new object. For example:
String s = "abc"; //create a new String object representing "abc"
s = s.toUpperCase(); //create another object representing "ABC"

Why is multiple inheritance not possible in Java?
It depends on how you understand "inheritance". Java can only "extends" one super class, but can "implements" many interfaces; that doesn't mean the multiple inheritance is not possible. You may use interfaces to make inheritance work for you. Or you may need to work around. For example, if you cannot get a feature from a class because your class has a super class already, you may get that class's feature by declaring it as a member field or getting an instance of that class. So the answer is that multiple inheritance in Java is possible.

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 relationship between synchronized and volatile keyword?
The JVM is guaranteed to treat reads and writes of data of 32 bits or less as atomic.(Some JVM might treat reads and writes of data of 64 bits or less as atomic in future) For long or double variable, programmers should take care in multi-threading environment. Either put these variables in a synchronized method or block, or declare them volatile.

This class (IncrementImpl) will be used by various threads concurrently; can you see the inherent flaw(s)? How would you improve it?
```
public class IncrementImpl {
private static int counter = 0;
public synchronized void increment() {
counter++;
}
public int getCounter() {
return counter;
}
}
```

The counter is static variable which is shared by multiple instances of this class. The increment() method is synchronized, but the getCounter() should be synchronized too. Otherwise the Java run-time system will not guarantee the data integrity and the race conditions will occur. The famous producer/consumer example listed at Sun's thread tutorial site will tell more.
one of solutions
```
public class IncrementImpl {
private static int counter = 0;
public synchronized void increment() {
counter++;
}
public synchronized int getCounter() {
return counter;
}
}
```

What are the drawbacks of inheritance?
Since inheritance inherits everything from the super class and interface, it may make the subclass too clustering and sometimes error-prone when dynamic overriding or dynamic overloading in some situation. In addition, the inheritance may make peers hardly understand your code if they don't know how your super-class acts and add learning curve to the process of development.
Usually, when you want to use a functionality of a class, you may use subclass to inherit such function or use an instance of this class in your class. Which is better, depends on your specification.

Is there any other way that you can achieve inheritance in Java?
There are a couple of ways. As you know, the straight way is to "extends" and/or "implements". The other way is to get an instance of the class to achieve the inheritance. That means to make the supposed-super-class be a field member. When you use an instance of the class, actually you get every function available from this class, but you may lose the dynamic features of OOP

Two methods have key words static synchronized and synchronized separately. What is the difference between them?
Both are synchronized methods. One is instance method, the other is class method. Method with static modifier is a class method. That means the method belongs to class itself and can be accessed directly with class name and is also called Singleton design. The method without static modifier is an instance method. That means the instance method belongs to its object. Every instance of the class gets its own copy of its instance method. When synchronized is used with a static method, a lock for the entire class is obtained. When synchronized is used with a non-static method, a lock for the particular object (that means instance) of the class is obtained. Since both methods are synchronized methods, you are not asked to explain what is a synchronized method. You are asked to tell the difference between instance and class method. Of course, your explanation to how synchronized keyword works doesn't hurt. And you may use this opportunity to show your knowledge scope.

How do you create a read-only collection?
The Collections class has six methods to help out here:
1. unmodifiableCollection(Collection c)
2. unmodifiableList(List list)
3. unmodifiableMap(Map m)
4. unmodifiableSet(Set s)
5. unmodifiableSortedMap(SortedMap m)
6. unmodifiableSortedSet(SortedSet s)
If you get an Iterator from one of these unmodifiable collections, when you call remove(), it will throw an UnsupportedOperationException.

Can a private method of a superclass be declared within a subclass?
Sure. A private field or method or inner class belongs to its declared class and hides from its subclasses. There is no way for private stuff to have a runtime overloading or overriding (polymorphism) features.

Why Java does not support multiple inheritance ?
This is a classic question. Yes or No depends on how you look at Java. If you focus on the syntax of "extends" and compare with C++, you may answer 'No' and give explanation to support you. Or you may answer 'Yes'. Recommend you to say 'Yes'.
Java DOES support multiple inheritance via interface implementation. Some people may not think in this way. Give explanation to support your point.

What is the difference between final, finally and finalize?
Short answer:
final - declares constant
finally - relates with exception handling
finalize - helps in garbage collection
If asked to give details, explain:

final field, final method, final class
try/finally, try/catch/finally
protected void finalize() in Object class

What kind of security tools are available in J2SE 5.0?
There are three tools that can be used to protect application working within the scope of security policies set at remote sites.
keytool -- used to manage keystores and certificates.
jarsigner -- used to generate and verify JAR signatures.
policytool -- used for managing policy files.
There are three tools that help obtain, list and manage Kerberos tickets.
kinit -- used to obtain Kerberos V5 tickets.
tklist -- used to list entries in credential cache and key tab.
ktab -- used to help manage entries in the key table.

How to make an array copy from System?
There is a method called arraycopy in the System class. You can do it:
System.arraycopy(sourceArray, srcOffset, destinationArray, destOffset, numOfElements2Copy);
When you use this method, the destinationArray will be filled with the elements of sourceArray at the length specified.

Can we use System.arraycopy() method to copy the same array?
Yes, you can. The source and destination arrays can be the same if you want to copy a subset of the array to another area within that array.

What is shallow copy or shallow clone in array cloning?
Cloning an array invloves creating a new array of the same size and type and copying all the old elements into the new array. But such copy is called shallow copy or shallow clone because any changes to the object would be reflected in both arrays.

When is the ArrayStoreException thrown?
When copying elements between different arrays, if the source or destination arguments are not arrays or their types are not compatible, an ArrayStoreException will be thrown.

How to check two arrays to see if contents have the same types and contain the same elements?
One of options is to use the equals() method of Arrays class.
Arrays.equals(a, b);
If the array types are different, a compile-time error will happen.

Can you call one constructor from another if a class has multiple constructors?
Yes. Use this() syntax.

What are the different types of inner classes?
There are four different types of inner classes in Java. They are: a)Static member classes , a static member class has access to all static methods of the parent, or top-level, class b) Member classes, the member class is instance specific and has access to any and all methods and members, even the parent's this reference c) Local classes, are declared within a block of code and are visible only within that block, just as any other method variable. d) Anonymous classes, is a local class that has no name

In which case would you choose a static inner class?
Interesting one, static inner classes can access the outer class's protected and private fields. This is both a positive and a negative point for us since we can, in essence, violate the encapsulation of the outer class by mucking up the outer class's protected and private fields. The only proper use of that capability is to write white-box tests of the class -- since we can induce cases that might be very hard to induce via normal black-box

tests (which don't have access to the internal state of the object). Second advantage,if I can say, is that, we can this static concept to impose restriction on the inner class. Again as discussed in earlier point, an Inner class has access to all the public, private and protected members of the parent class. Suppose you want to restrict the access even to inner class, how would you go ahead? Making the inner class static enforces it to access only the public static members of the outer class(Since, protected and private members are not supposed to be static and that static members can access only other static members). If it has to access any non-static member, it has to create an instance of the outer class which leads to accessing only public members.

What is weak reference in Java
A weak reference is one that does not prevent the referenced object from being garbage collected. You might use them to manage a HashMap to look up a cache of objects. A weak reference is a reference that does not keep the object it refers to alive. A weak reference is not counted as a reference in garbage collection. If the object is not referred to elsewhere as well, it will be garbage collected.

What is the difference between final, finally and finalize?
final is used for making a class no-subclassable, and making a member variable as a constant which cannot be modified. finally is usually used to release all the resources utilized inside the try block. All the resources present in the finalize method will be garbage collected whenever GC is called. Though finally and finalize seem to be for a similar task there is an interesting tweak here, usually I prefer finally than finalize unless it is unavoidable. This is because the code in finally block is guaranteed of execution irrespective of occurrence of exception, while execution of finalize is not guarenteed.finalize method is called by the garbage collector on an object when the garbage collector determines that there are no more references to the object. Presumably the garbage collector will, like its civil servant namesake, visit the heap on a regular basis to clean up resources that are no longer in use. Garbage collection exists to prevent programmers from calling delete. This is a wonderful feature. For example, if you can't call delete, then you can't accidentally call delete twice on the same object. However, removing delete from the language is not the same thing as automatically cleaning up. To add to it, Garbage collection might not ever run. If garbage collection runs at all, and an object is no longer referenced, then that object's finalize will run. Also, across multiple objects, finalize order is not predictable. The correct approach to resource cleanup in Java language programs does not rely on finalize. Instead, you simply write explicit close methods for objects that wrap native resources. If you take this approach, you must document that the close method exists and when it should be called. Callers of the object must then remember to call close when they are finished with a resource.

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.

The following statement prints true or false, why?
```
byte[] a = { 1, 2, 3 };,
byte[] b = (byte[]) a.clone();
System.out.println(a == b);
```
The false will be printed out. Because the two arrays have distinctive memory addresses. Starting in Java 1.2, we can use java.util.Arrays.equals(a, b) to compare whether two arrays have the same contents.

Why do we need to use getSystemResource() and getSystemResources() method to load resources?
Because we want to look for resources strictly from the system classpath, These methods use the system ClassLoader to locate resources, which gives you stricter control of the resources used by the application.

ArithmeticException?
The ArithmeticException is thrown when integer is divided by zero or taking the remainder of a number by zero. It is never thrown in floating-point operations.

What is a transient variable?
A transient variable is a variable that may not be serialized.

Which containers use a border Layout as their default layout?
The window, Frame and Dialog classes use a border layout as their default layout.

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 the output from System.out.println("Hello"+null);?
Hellonull

What is synchronization and why is it important?
With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

Can 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's new with the stop(), suspend() and resume() methods in JDK 1.2?
The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

Is null a keyword?
The null value is not a keyword.

What is the preferred size of a component?
The preferred size of a component is the minimum component size that will allow the component to display normally.

What method is used to specify a container's layout?
The setLayout() method is used to specify a container's layout.

Which containers use a FlowLayout as their default layout?
The Panel and Applet classes use the FlowLayout as their default layout.

What state does a thread enter when it terminates its processing?
When a thread terminates its processing, it enters the dead state.

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

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 is the List interface?
The List interface provides support for ordered collections of objects.

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

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.

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

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.

Which method of the Component class is used to set the position and size of a component?
setBounds()

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 is the difference between yielding and sleeping?
When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

Which java.util classes and interfaces support event handling?
The EventObject class and the EventListener interface support event processing.

Is sizeof a keyword?
The sizeof operator is not a keyword.

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

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 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).

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 is the immediate superclass of the Applet class?
Panel

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.

Name three Component subclasses that support painting.
The Canvas, Frame, Panel, and Applet classes support painting.

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.

What is the immediate superclass of the Dialog class?
Window.

What is clipping?
Clipping is the process of confining paint operations to a limited area or shape.

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

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

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. Associatity determines whether an expression is evaluated left-to-right or right-to-left

When a thread blocks on I/O, what state does it enter?
A thread enters the waiting state when it blocks on I/O.

To what value is a variable of the String type automatically initialized?
The default value of a String type is null.

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.

What is the difference between a MenuItem and a CheckboxMenuItem?
The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.

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 class is the top of the AWT event hierarchy?
The java.awt.AWTEvent class is the highest-level class in the AWT event-class hierarchy.

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.

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

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

In which package are most of the AWT events that support the event-delegation model defined?
Most of the AWT-related events of the event-delegation model are defined in the java.awt.event package. The AWTEvent class is defined in the java.awt package.

What is the immediate super class of Menu?
What is the immediate super class of Menu? MenuItem

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.

Which class is the immediate super class of the MenuComponent class.
Object

What invokes a thread's run() method?
After a thread is started, via its start() method or that of the Thread class, the JVM invokes the thread's run() method when the thread is initially executed.

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.

Name three subclasses of the Component class.
Box.Filler, Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, or TextComponent

What is the GregorianCalendar class?
The GregorianCalendar provides support for traditional Western calendars.

Which Container method is used to cause a container to be laid out and redisplayed?
validate()

What is the purpose of the Runtime class?
The purpose of the Runtime class is to provide access to the Java runtime system.

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? garbage collector?
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.

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.

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 method is invoked to cause an object to begin executing as a separate thread?
The start() method of the Thread class is invoked to cause an object to begin executing as a separate thread.

Name two subclasses of the TextComponent class.
TextField and TextArea

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.

Which containers may have a MenuBar?
Frame

How are commas used in the initialization 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 the purpose of the wait(), notify(), and notifyAll() methods?
The wait(),notify(), and notifyAll() methods are used to provide an efficient way for threads to wait for a shared resource. When a thread executes an object's wait() method, it enters the waiting state. It only enters the ready state after another thread invokes the object's notify() or notifyAll() methods.

What is an abstract method?
An abstract method is a method whose implementation is deferred to a subclass.

How are Java source code files named?
A Java source code file takes the name of a public class or interface that is defined within the file. A source code file may contain at most one public class or interface. If a public class or interface is defined within a source code file, then the source code file must take the name of the public class or interface. If no public class or interface is defined within a source code file, then the file must take on a name that is different than its classes and interfaces. Source code files use the .java extension.

What is the relationship between the Canvas class and the Graphics class?
A Canvas object provides access to a Graphics object via its paint() method.

What are the high-level thread states?
The high-level thread states are ready, running, waiting, and dead.

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.

What is the difference between the String and StringBuffer classes?
String objects are constants. StringBuffer objects are not.

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 Dictionary class?
The Dictionary class provides the capability to store key-value pairs.

How are the elements of a BorderLayout organized?
The elements of a BorderLayout are organized at the borders (North, South, East, and West) and the center of a container.

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.

What is the difference between a Window and a Frame?
The Frame class extends Window to define a main application window that can have a menu bar.

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.

What is the difference between the Font and FontMetrics classes?
The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object.

How is rounding performed under integer division?
The fractional part of the result is truncated. This is known as rounding toward zero.

What happens when a thread cannot acquire a lock on an object?
If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

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.

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.

Does a class inherit the constructors of its superclass?
A class does not inherit constructors from any of its super classes.

For which statements does it make sense to use a label?
The only statements for which it makes sense to use a label are those statements that can enclose a break or continue statement.

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

Which TextComponent method is used to set a TextComponent to the read-only state?
setEditable()

How are the elements of a CardLayout organized?
The elements of a CardLayout are stacked, one on top of the other, like a deck of cards.

Is &&= a valid Java operator?
No, it is not.

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.

What is the relationship between clipping and repainting?
When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.

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

What is the relationship between an event-listener interface and an event-adapter class?
An event-listener interface defines the methods that must be implemented by an event handler for a particular kind of event. An event adapter provides a default implementation of an event-listener interface.

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 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 event results from the clicking of a button?
The ActionEvent event is generated as the result of the clicking of a button.

How can a GUI component handle its own events?
A component can handle its own events by implementing the required event-listener interface and adding itself as its own event listener.

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.

How are the elements of a GridBagLayout organized?
The elements of a GridBagLayout are organized according to a grid. However, the elements are of different sizes and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

What advantage do Java's layout managers provide over traditional windowing systems?
Java uses layout managers to lay out components in a consistent manner across all windowing platforms. Since Java's layout managers aren't tied to absolute sizing and positioning, they are able to accommodate platform-specific differences among windowing systems.

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 modifiers can be used with a local inner class?
A local inner class may be final or abstract.

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.

What is the difference between the paint() and repaint() methods?
The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.

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.

Which Math method is used to calculate the absolute value of a number?
The abs() method is used to calculate absolute values.

How does multithreading take place on a computer with a single CPU?
The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

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.

When is the finally clause of a try-catch-finally statement executed?
The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

Which class is the immediate superclass of the Container class?
Component

If a method is declared as protected, where may the method be 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.

How can the Checkbox class be used to create a radio button?
By associating Checkbox objects with a CheckboxGroup.

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 happens when you invoke a thread's interrupt method while it is sleeping or waiting?
When a task's interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.

What is the return type of a program's main() method?
A program's main() method has a void return type.

Name four Container classes.
Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

What is the difference between a Choice and a List?
A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices. Only one item may be selected from a Choice. A List may be displayed in such a way that several List items are visible. A List supports the selection of one or more List items.

What class of exceptions are 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.

Under what conditions is an object's finalize() method invoked by the garbage collector?
The garbage collector invokes an object's finalize() method when it detects that the object has become unreachable.

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.

What is the difference between the JDK 1.02 event model and the event-delegation model introduced with JDK 1.1?
The JDK 1.02 event model uses an event inheritance or bubbling approach. In this model, components are required to handle their own events. If they do not handle a particular event, the event is inherited by (or bubbled up to) the component's container. The container then either handles the event or it is bubbled up to its container and so on, until the highest-level container has been tried. In the event-delegation model, specific objects are designated as event handlers for GUI components. These objects implement event-listener interfaces. The event-delegation model is more efficient than the event-inheritance model because it eliminates the processing required to support the bubbling of unhandled events.

How is it possible for two String objects with identical values not to be equal under the == operator?
The == operator compares two objects to determine if they are the same object in memory. It is possible for two String objects to have the same value, but located indifferent areas of memory.

Why are the methods of the Math class static?
So they can be invoked as if they are a mathematical code library.

What Checkbox method allows you to tell if a Checkbox is checked?
getState()

What state is a thread in when it is executing?
An executing thread is in the running state.

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.

How are the elements of a GridLayout organized?
The elements of a GridBad layout are of equal size and are laid out using the squares of a grid.

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 a void return type?
A void return type indicates that a method does not return a value.

What is the purpose of the enableEvents() method?
The enableEvents() method is used to enable an event for a particular object. Normally, an event is enabled when a listener is added to an object for a particular event. The enableEvents() method is used by objects that handle events by overriding their event-dispatch methods.

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 are this and super used?
this is used to refer to the current object instance. super is used to refer to the variables and methods of the superclass of the current object instance.

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 is a compilation unit?
A compilation unit is a Java source code file.

What interface is extended by AWT event listeners?
All AWT event listeners extend the java.util.EventListener interface.

What restrictions are placed on method overriding?
Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden method.

How can a dead thread be restarted?
A dead thread cannot be restarted.

What happens if an exception is not caught?
An uncaught exception results in the uncaughtException() method of the thread's ThreadGroup being invoked, which eventually results in the termination of the program in which it is thrown.

What is a layout manager?
A layout manager is an object that is used to organize components in a container.

Which arithmetic operations can result in the throwing of an ArithmeticException?
Integer / and % can result in the throwing of an ArithmeticException.

What are three ways in which a thread can enter the waiting state?
A thread can enter the waiting state by invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

Can an abstract class be final?
An abstract class may not be declared as final.

What is the ResourceBundle class?
The ResourceBundle class is used to store locale-specific resources that can be loaded by a program to tailor the program's appearance to the particular locale in which it is being run.

What happens if a try-catch-finally statement does not have a catch clause to handle an exception that is thrown within the body of the try statement?
The exception propagates up to the next higher level try-catch statement (if any) or results in the program's termination.

What is numeric promotion?
Numeric promotion is the conversion of a smaller numeric type to a larger numeric type, so that integer and floating-point operations may take place. In numerical promotion, byte, char, and short values are converted to int values. The int values are also converted to long values, if necessary. The long and float values are converted to double values, as required.

What is the difference between a Scrollbar and a ScrollPane?
A Scrollbar is a Component, but not a Container. A ScrollPane is a Container. A ScrollPane handles its own events and performs its own scrolling.

What is the difference between a public and a non-public class?
A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

To what value is a variable of the boolean type automatically initialized?
The default value of the boolean type is false.

Can try statements be nested?
Try statements may be tested.

What is the difference between the prefix and postfix forms of the ++ operator?
The prefix form performs the increment operation and returns the value of the increment operation. The postfix form returns the current value all of the expression and then performs the increment operation on that value.

What is the purpose of a statement block?
A statement block is used to organize a sequence of statements as a single statement group.

What is a Java package and how is it used?
A Java package is a naming context for classes and interfaces. A package is used to create a separate name space for groups of classes and interfaces. Packages are also used to organize related classes and interfaces into a single API unit and to control accessibility to these classes and interfaces.

What modifiers may be used with a top-level class?
A top-level class may be public, abstract, or final.

What are the Object and Class classes used for?
The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program.

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.

When is an object subject to garbage collection?
An object is subject to garbage collection when it becomes unreachable to the program in which it is used.

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 methods are used to get and set the text label displayed by a Button object?
getLabel() and setLabel()

Which Component subclass is used for drawing and painting?
Canvas

What are the two basic ways in which classes that can be run as threads may be defined?
A thread class may be declared as a subclass of Thread, or it may implement the Runnable interface.

What are the problems faced by Java programmers who don't use layout managers?
Without layout managers, Java programmers are faced with determining how their GUI will be displayed across multiple windowing systems and finding a common sizing and positioning that will work within the constraints imposed by each windowing system.

What is the difference between an if statement and a switch statement?
The if statement is used to select among two alternatives. It uses a Boolean expression to decide which alternative should be executed. The switch statement is used to select among multiple alternatives. It uses an int expression to determine which alternative should be executed.

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

Can an Interface be final?
yes.

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("in interfia"); }; public static void main(String a1[]) { System.out.println("in interfia"); } } }

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

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

I made my class Cloneable but I still get 'Can't access protected method clone. Why?
Yeah, some of the Java books, in particular "The Java Programming Language", 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().

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.

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 is 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 the main difference between Java platform and other platforms?
The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.
The Java platform has three elements:
Java programming language
The Java Virtual Machine (Java VM)
The Java Application Programming Interface (Java API)

What is the Java Virtual Machine?
The Java Virtual Machine is a software that can be ported onto various hardware-based platforms.

What is the Java API?
The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.

What is the package?
The package is a Java namespace or part of Java libraries. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

What is native code?
The native code is code that after you compile it, the compiled code runs on a specific hardware platform.

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
}
```

Is Java code slower than native code?
Not really. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

Can main() method be overloaded?
Yes. the main() method is a special method for a program entry. You can overload main() method in any ways. But if you change the signature of the main method, the entry point for the program will be gone.

What is the serialization?
The serialization is a kind of mechanism that makes a class or a bean persistence by having its properties or fields and state information saved and restored to and from storage.

Explain the new Features of JDBC 2.0 Core API?
The JDBC 2.0 API includes the complete JDBC API, which includes both core and Optional Package API, and provides inductrial-strength database computing capabilities.
New Features in JDBC 2.0 Core API:

Scrollable result sets- using new methods in the ResultSet interface allows programmatically move the to particular row or to a position relative to its current position
JDBC 2.0 Core API provides the Batch Updates functionality to the java applications.
Java applications can now use the ResultSet.updateXXX methods.
New data types - interfaces mapping the SQL3 data types
Custom mapping of user-defined types (UTDs)
Miscellaneous features, including performance hints, the use of character streams, full precision for java.math.BigDecimal values, additional security, and support for time zones in date, time, and timestamp values.

How you can force the garbage collection?
Garbage collection automatic process and can't be forced.

Explain garbage collection?
Garbage collection is one of the most important feature of Java. Garbage collection is also called automatic memory management as JVM automatically removes the unused variables/objects (value is null) from the memory. User program cann't directly free the object from memory, instead it is the job of the garbage collector to automatically free the objects that are no longer referenced by a program. Every class inherits finalize() method from java.lang.Object, the finalize() method is called by garbage collector when it determines no more references to the object exists. In Java, it is good idea to explicitly assign null into a variable when no more in use. I Java on calling System.gc() and Runtime.gc(), JVM tries to recycle the unused objects, but there is no guarantee when all the objects will garbage collected.

Describe the principles of OOPS.
There are three main principals of oops which are called Polymorphism, Inheritance and Encapsulation.

Explain the Encapsulation principle.
Encapsulation is a process of binding or wrapping the data and the codes that operates on the data into a single entity. This keeps the data safe from outside interface and misuse. One way to think about encapsulation is as a protective wrapper that prevents code and data from being arbitrarily accessed by other code defined outside the wrapper.

Explain the Inheritance principle.
Inheritance is the process by which one object acquires the properties of another object.

Explain the Polymorphism principle.
The meaning of Polymorphism is something like one name many forms. Polymorphism enables one entity to be used as as general category for different types of actions. The specific action is determined by the exact nature of the situation. The concept of polymorphism can be explained as "one interface, multiple methods".

Explain the different forms of Polymorphism.
From a practical programming viewpoint, polymorphism exists in three distinct forms in Java:
Method overloading
Method overriding through inheritance
Method overriding through the Java interface

What are Access Specifiers available in Java?
ccess specifiers are keywords that determines the type of access to the member of a class. These are:
Public
Protected
Private
Defaults

Describe the wrapper classes in Java.
Wrapper class is wrapper around a primitive data type. An instance of a wrapper class contains, or wraps, a primitive value of the corresponding type.

Following table lists the primitive types and the corresponding wrapper classes:

Primitive Wrapper
boolean java.lang.Boolean
byte java.lang.Byte
char java.lang.Character
double java.lang.Double
float java.lang.Float
int java.lang.Integer
long java.lang.Long
short java.lang.Short
void java.lang.Void

Question: Read the following program:
```
public class test {
public static void main(String [] args) {
int x = 3;
int y = 1;
if (x = y)
System.out.println("Not equal");
else
System.out.println("Equal");
}
}
```

What is the result?
A. The output is “Equal”
B. The output in “Not Equal”
C. An error at " if (x = y)" causes compilation to fall.
D. The program executes but no output is show on console.

Answer: C

Use the Externalizable interface when you need complete control over your Bean's serialization (for example, when writing and reading a specific file format).
No. Earlier order is maintained.

The superclass constructor runs before the subclass constructor. The subclass's version of the overridable method will be invoked before the subclass's constructor has been invoked. If the subclass's overridable method depends on the proper initialization of the subclass (through the subclass constructor), the method will most likely fail. Is that true?
Yes. It is true

Why are the interfaces more flexible than abstract classes?
--An interface-defined type can be implemented by any class in a class hierarchy and can be extended by another interface. In contrast, an abstract-class-defined type can be implemented only by classes that subclass the abstract class.
--An interface-defined type can be used well in polymorphism. The so-called interface type vs. implementation types.
--Abstract classes evolve more easily than interfaces. If you add a new concrete method to an abstract class, the

hierarchy system is still working. If you add a method to an interface, the classes that rely on the interface will break when recompiled.
--Generally, use interfaces for flexibility; use abstract classes for ease of evolution (like expanding class functionality).

What are new language features in J2SE 5.0?
Generally:
1. generics
2. static imports
3. annotations
4. typesafe enums
5. enhanced for loop
6. autoboxing/unboxing
7. varargs
8. covariant return types

What is covariant return type?
A covariant return type lets you override a superclass method with a return type that subtypes the superclass method's return type. So we can use covariant return types to minimize upcasting and downcasting.

```
class Parent {
Parent foo () {
System.out.println ("Parent foo() called");
return this;
}
}

class Child extends Parent {
Child foo () {
System.out.println ("Child foo() called");
return this;
}
}

class Covariant {
public static void main(String[] args) {
Child c = new Child();
Child c2 = c.foo(); // c2 is Child
Parent c3 = c.foo(); // c3 points to Child
}
}
```

What is the result of the following statement?
int i = 1, float f = 2.0f;
i += f; //ok, the cast done automatically by the compiler
i = i + f; //error
The compound assignment operators automatically include cast operations in their behaviors.

What is externalization? Where is it useful?
Use the Externalizable interface when you need complete control over your Bean's serialization (for example, when writing and reading a specific file format).

What will be the output on executing the following code.

```
public class MyClass {
public static void main (String args[] ) {
```

int abc[] = new int [5];
System.out.println(abc);
}
}

A Error array not initialized
B 5
C null
D Print some junk characters

Answer : D

It will print some junk characters to the output. Here it will not give any compile time or runtime error because we have declared and initialized the array properly. Event if we are not assigning a value to the array, it will always initialized to its defaults.

What will be the output on executing the following code.
public class MyClass {
public static void main (String args[]) {
int abc[] = new int [5];
System.out.println(abc[0]);
}
}
A Error array not initialized
B 5
C 0
D Print some junk characters

Answer : C.

What is a marker interface ?
An interface that contains no methods. E.g.: Serializable, Cloneable, SingleThreadModel etc. It is used to just mark java classes that support certain capability.

What are tag interfaces?
Tag interface is an alternate name for marker interface.

What are the restrictions placed on static method ?
We cannot override static methods. We cannot access any object variables inside static method. Also the this reference also not available in static methods.

What is JVM?
JVM stands for Java Virtual Machine. It is the run time for java programs. All are java programs are running inside this JVM only. It converts java byte code to OS specific commands. In addition to governing the execution of an application's byte codes, the virtual machine handles related tasks such as managing the system's memory, providing security against malicious code, and managing multiple threads of program execution.

What is JIT?
JIT stands for Just In Time compiler. It compiles java byte code to native code.

What are ClassLoaders?
A class loader is an object that is responsible for loading classes. The class ClassLoader is an abstract class. Given the name of a class, a class loader should attempt to locate or generate data that constitutes a definition

for the class. A typical strategy is to transform the name into a file name and then read a "class file" of that name from a file system.
Every Class object contains a reference to the ClassLoader that defined it.
Class objects for array classes are not created by class loaders, but are created automatically as required by the Java runtime. The class loader for an array class, as returned by Class.getClassLoader() is the same as the class loader for its element type; if the element type is a primitive type, then the array class has no class loader.
Applications implement subclasses of ClassLoader in order to extend the manner in which the Java virtual machine dynamically loads classes.

What is Service Locator pattern?
The Service Locator pattern locates J2EE (Java 2 Platform, Enterprise Edition) services for clients and thus abstracts the complexity of network operation and J2EE service lookup as EJB (Enterprise JavaBean) Interview Questions - Home and JMS (Java Message Service) component factories. The Service Locator hides the lookup process's implementation details and complexity from clients. To improve application performance, Service Locator caches service objects to eliminate unnecessary JNDI (Java Naming and Directory Interface) activity that occurs in a lookup operation.

What is Session Facade pattern?
Session facade is one design pattern that is often used while developing enterprise applications. It is implemented as a higher level component (i.e.: Session EJB), and it contains all the iteractions between low level components (i.e.: Entity EJB). It then provides a single interface for the functionality of an application or part of it, and it decouples lower level components simplifying the design. Think of a bank situation, where you have someone that would like to transfer money from one account to another. In this type of scenario, the client has to check that the user is authorized, get the status of the two accounts, check that there are enough money on the first one, and then call the transfer. The entire transfer has to be done in a single transaction otherwise is something goes south, the situation has to be restored.
As you can see, multiple server-side objects need to be accessed and possibly modified. Multiple fine-grained invocations of Entity (or even Session) Beans add the overhead of network calls, even multiple transaction. In other words, the risk is to have a solution that has a high network overhead, high coupling, poor reusability and mantainability.
The best solution is then to wrap all the calls inside a Session Bean, so the clients will have a single point to access (that is the session bean) that will take care of handling all the rest.

What is Data Access Object pattern?
The Data Access Object (or DAO) pattern: separates a data resource's client interface from its data access mechanisms adapts a specific data resource's access API to a generic client interface
The DAO pattern allows data access mechanisms to change independently of the code that uses the data.
The DAO implements the access mechanism required to work with the data source. The data source could be a persistent store like an RDBMS, an external service like a B2B exchange, a repository like an LDAP database, or a business service accessed via CORBA Internet Inter-ORB Protocol (IIOP) or low-level sockets. The business component that relies on the DAO uses the simpler interface exposed by the DAO for its clients. The DAO completely hides the data source implementation details from its clients. Because the interface exposed by the DAO to clients does not change when the underlying data source implementation changes, this pattern allows the DAO to adapt to different storage schemes without affecting its clients or business components. Essentially, the DAO acts as an adapter between the component and the data source.

Can we make an EJB singleton?
This is a debatable question, and for every answer we propose there can be contradictions. I propose 2 solutions of the same. Remember that EJB's are distributed components and can be deployed on different JVM's in a Distributed environment
i) Follow the steps as given below
Make sure that your serviceLocator is deployed on only one JVM.
In the serviceLocator create a HashTable/HashMap(You are the right judge to choose between these two)
When ever a request comes for an EJB to a serviceLocator, it first checks in the HashTable if an entry already

exists in the table with key being the JNDI name of EJB. If key is present and value is not null, return the existing reference, else lookup the EJB in JNDI as we do normally and add an entry into the Hashtable before returning it to the client. This makes sure that you maintain a singleton of EJB.
ii) In distributed environment our components/Java Objects would be running on different JVM's. So the normal singleton code we write for maintaining single instance works fine for single JVM, but when the class could be loaded in multiple JVM's and Instantiated in multiple JVM's normal singleton code does not work. This is because the ClassLoaders being used in the different JVM's are different from each other and there is no defined mechanism to check and compare what is loaded in another JVM. A solution could be(Not tested yet. Need your feedback on this) to write our own ClassLoader and pass this classLoader as argument, whenever we are creating a new Instance and make sure that only one instance is created for the proposed class. This can be done easily.

How can we make a class Singleton ?
A) If the class is Serializable

```
class Singleton implements Serializable
{
private static Singleton instance;

private Singleton() { }

public static synchronized Singleton getInstance()
{
if (instance == null)
instance = new Singleton();
return instance;
}

/**
If the singleton implements Serializable, then this
* method must be supplied.
*/
protected Object readResolve() {
return instance;
}

/**
This method avoids the object fro being cloned
*/
public Object clone() {
throws CloneNotSupportedException ;
//return instance;
}
}
```

B) If the class is NOT Serializable

```
class Singleton
{
private static Singleton instance;
private Singleton() { }

public static synchronized Singleton getInstance()
{
```

```
if (instance == null)
instance = new Singleton();
return instance;
}

/**
This method avoids the object from being cloned
**/
public Object clone() {
throws CloneNotSupportedException ;
//return instance;
}

}
```

How is static Synchronization different form non-static synchronization?
When Synchronization is applied on a static Member or a static block, the lock is performed on the Class and not on the Object, while in the case of a Non-static block/member, lock is applied on the Object and not on class. [Trail 2: There is a class called Class in Java whose object is associated with the object(s) of your class. All the static members declared in your class will have reference in this class(Class). As long as your class exists in memory this object of Class is also present. Thats how even if you create multiple objects of your class only one Class object is present and all your objects are linked to this Class object. Even though one of your object is GCed after some time, this object of Class is not GCed untill all the objects associated with it are GCed.
This means that when ever you call a "static synchronized" block, JVM locks access to this Class object and not any of your objects. Your client can till access the non-static members of your objects.

What are class members and Instance members?
Any global members(Variables, methods etc.) which are static are called as Class level members and those which are non-static are called as Instance level members.

Name few Garbage collection algorithms?
Here they go:
Mark and Sweep
Reference counting
Tracing collectors
Copying collectors
Heap compaction
Mark-compact collectors

Can we force Garbage collection?
java follows a philosophy of automatic garbage collection, you can suggest or encourage the JVM to perform garbage collection but you can not force it. Once a variable is no longer referenced by anything it is available for garbage collection. You can suggest garbage collection with System.gc(), but this does not guarantee when it will happen. Local variables in methods go out of scope when the method exits. At this point the methods are eligible for garbage collection. Each time the method comes into scope the local variables are re-created.

What are generations in Garbage Collection terminology? What is its relevance?
Garbage Collectors make assumptions about how our application runs. Most common assumption is that an object is most likely to die shortly after it was created: called infant mortality. This assumes that an object that has been around for a while, will likely stay around for a while. GC organizes objects into generations (young, tenured, and perm). This tells that if an object lives for more than certain period of time it is moved from one generation to another generations(say from young -> tenured -> permanent). Hence GC will be run more

frequently at the young generations and rarely at permanent generations. This reduces the overhead on GC and gives faster response time.

What is a Throughput Collector?
The throughput collector is a generational collector similar to the default collector but with multiple threads used to do the minor collection. The major collections are essentially the same as with the default collector. By default on a host with N CPUs, the throughput collector uses N garbage collector threads in the collection. The number of garbage collector threads can be controlled with a command line option.

When to Use the Throughput Collector?
Use the throughput collector when you want to improve the performance of your application with larger numbers of processors. In the default collector garbage collection is done by one thread, and therefore garbage collection adds to the serial execution time of the application. The throughput collector uses multiple threads to execute a minor collection and so reduces the serial execution time of the application. A typical situation is one in which the application has a large number of threads allocating objects. In such an application it is often the case that a large young generation is needed

What is Aggressive Heap?
The -XX:+AggressiveHeap option inspects the machine resources (size of memory and number of processors) and attempts to set various parameters to be optimal for long-running, memory allocation-intensive jobs. It was originally intended for machines with large amounts of memory and a large number of CPUs, but in the J2SE platform, version 1.4.1 and later it has shown itself to be useful even on four processor machines. With this option the throughput collector (-XX:+UseParallelGC) is used along with adaptive sizing (-XX:+UseAdaptiveSizePolicy). The physical memory on the machines must be at least 256MB before Aggressive Heap can be used.

What is a Concurrent Low Pause Collector?
The concurrent low pause collector is a generational collector similar to the default collector. The tenured generation is collected concurrently with this collector. This collector attempts to reduce the pause times needed to collect the tenured generation. It uses a separate garbage collector thread to do parts of the major collection concurrently with the applications threads. The concurrent collector is enabled with the command line option -XX:+UseConcMarkSweepGC. For each major collection the concurrent collector will pause all the application threads for a brief period at the beginning of the collection and toward the middle of the collection. The second pause tends to be the longer of the two pauses and multiple threads are used to do the collection work during that pause. The remainder of the collection is done with a garbage collector thread that runs concurrently with the application. The minor collections are done in a manner similar to the default collector, and multiple threads can optionally be used to do the minor collection.

When to Use the Concurrent Low Pause Collector?
Use the concurrent low pause collector if your application would benefit from shorter garbage collector pauses and can afford to share processor resources with the garbage collector when the application is running. Typically applications which have a relatively large set of long-lived data (a large tenured generation), and run on machines with two or more processors tend to benefit from the use of this collector. However, this collector should be considered for any application with a low pause time requirement. Optimal results have been observed for interactive applications with tenured generations of a modest size on a single processor.

What is Incremental Low Pause Collector?
The incremental low pause collector is a generational collector similar to the default collector. The minor collections are done with the same young generation collector as the default collector. Do not use either -XX:+UseParallelGC or -XX:+UseParNewGC with this collector. The major collections are done incrementally on the tenured generation. This collector (also known as the train collector) collects portions of the tenured generation at each minor collection. The goal of the incremental collector is to avoid very long major collection pauses by doing portions of the major collection work at each minor collection. The incremental collector will

sometimes find that a non-incremental major collection (as is done in the default collector) is required in order to avoid running out of memory.

When to Use the Incremental Low Pause Collector?
Use the incremental low pause collector when your application can afford to trade longer and more frequent young generation garbage collection pauses for shorter tenured generation pauses. A typical situation is one in which a larger tenured generation is required (lots of long-lived objects), a smaller young generation will suffice (most objects are short-lived and don't survive the young generation collection), and only a single processor is available.

How do you enable the concurrent garbage collector on Sun's JVM?
-Xconcgc options allows us to use concurrent garbage collector (1.2.2_07+)we can also use -XX:+UseConcMarkSweepGC which is available beginning with J2SE 1.4.1.

What is a platform?
A platform is the hardware or software environment in which a program runs. Most platforms can be described as a combination of the operating system and hardware, like Windows 2000 and XP, Linux, Solaris, and MacOS.

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

How to make a class or a bean serializable?
By implementing either the java.io.Serializable interface, or the java.io.Externalizable interface. As long as one class in a class's inheritance hierarchy implements Serializable or Externalizable, that class is serializable.

What restrictions are placed on method overloading?
Two methods may not have the same name and argument list but different return types.

Name Container classes.
Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

What is the List interface?
The List interface provides support for ordered collections of objects.

What is the difference between a Scrollbar and a ScrollPane?
A Scrollbar is a Component, but not a Container. A ScrollPane is a Container. A ScrollPane handles its own events and performs its own scrolling.

What is tunnelling?
Tunnelling is a route to somewhere. For example, RMI tunnelling is a way to make RMI application get through firewall. In CS world, tunnelling means a way to transfer data.

What is meant by "Abstract Interface"?
First, an interface is abstract. That means you cannot have any implementation in an interface. All the methods declared in an interface are abstract methods or signatures of the methods.

Can Java code be compiled to machine dependent executable file?
Yes. There are many tools out there. If you did so, the generated exe file would be run in the specific platform, not cross-platform.

Do not use the String contatenation operator in lengthy loops or other places where performance could suffer. Is that true?
Yes.

What method is used to specify a container's layout?
The setLayout() method is used to specify a container's layout.

Which containers use a FlowLayout as their default layout?
The Panel and Applet classes use the FlowLayout as their default layout.

What state does a thread enter when it terminates its processing?
When a thread terminates its processing, it enters the dead state.

What is 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.

Is sizeof a keyword?
The sizeof operator is not a keyword in Java.

Which class is the superclass for every class.
Object.

Which Container method is used to cause a container to be laid out and redisplayed?
validate()

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

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

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 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 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..the only interface can help.

What are pass by reference and passby value?
Pass By Reference means the passing the address itself rather than passing the value. Passby 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.

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 doesnt allow). HashMap does not guarantee that the order of the map will remain constant over time. HashMap is unsynchronized and Hashtable is synchronized.

Difference between Vector and ArrayList?
Vector is synchronized whereas arraylist is not.

Difference between Swing and AWT?
AWT are heavy-weight components. Swings are light-weight components. Hence swing works faster than AWT.

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 ie, without any access modifier (ie, 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 (ie, 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 super class 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 no static 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 ie., 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.

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 abt 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. eg, 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. eg, 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, Member classes, Local classes, Anonymous 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. eg, 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 a 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 serilized.

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 alongwith the original obect.

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 doesnot necessarily read and write to the stream. These are
1. Serialization ignores static fields, because they are not part of ay particular state state.
2. Base class fields are only hendled if the base class itself is serializable.
3. 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.

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, Double etc.

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.

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 are 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,
1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions. and
2. 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.
1. try catch block and
2. 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 yourself 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 creators 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 is daemon thread and which method is used to create the daemon thread?
Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

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 classs 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 : Exceute 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 exceptionis 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 contain 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 NullPointerException 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.
1. 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.
2. Local : - These are the variables that are defined within a method. They remain accessible only during the course of method execution. When the method finishes execution, these variables fall out of scope.
3. 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.