OBJECT ORIENTED PROGRAMMING WITH JAVA(OWJ)

Question: Can a private method of a superclass be declared within a subclass?
Answer: 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.

Question: Why Java does not support multiple inheritence ?
Answer: Java DOES support multiple inheritance via interface implementation.

Question: What is the difference between final, finally and finalize?
Answer: o final - declare constant
o finally - handles exception
o finalize - helps in garbage collection

Question: Where and how can you use a private constructor?
Answer: Private constructor can be used if you do not want any other class to instanstiate the object, the instantiation is done from a static public method, this method is used when dealing with the factory method pattern when the designer wants only one controller (fatory method) to create the object.

Question: In System.out.println(),what is System,out and println,pls explain?
Answer: *System* is a predefined final class,*out* is a PrintStream object and *println* is a built-in overloaded method in the *out* object.

Question: What is meant by "Abstract Interface"?
Answer: 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.

Question: 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.
Answer: 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.

Question: What is the output of x<y? a:b = p*q when x=1,y=2,p=3,q=4?
Answer: When this kind of question has been asked, find the problems you think is necessary to ask 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<y? a:(b=p*q); the return value would be variable a because the x is 1 and less than y = 2; the x < y statement return true and variable a is returned.

Question: What is the difference between Swing and AWT components?
Answer: 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.

Question: Why Java does not support pointers?
Answer: 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# shine.

Question: What is a platform?
Answer: 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/XP, Linux, Solaris, and MacOS.

Question: What is the main difference between Java platform and other platforms?
Answer: 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 two components:

1. The Java Virtual Machine (Java VM)

2. The Java Application Programming Interface (Java API)

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

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

Question: What is the package?
Answer: 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.

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

Question: Is Java code slower than native code?
Answer: 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.

Question: What is the serialization?
Answer: 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.

Question: How to make a class or a bean serializable?
Answer: 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.

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

Question: What is synchronization and why is it important?
Answer: 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.

Question: What are synchronized methods and synchronized statements?
Answer: 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.

Question: What is synchronization and why is it important?
Answer: 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.

Question: What are synchronized methods and synchronized statements?
Answer: 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.

Question: What are three ways in which a thread can enter the waiting state?
Answer: 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.

Question: Can a lock be acquired on a class?
Answer: Yes, a lock can be acquired on a class. This lock is acquired on the class's Class object.

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

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

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

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

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

Question: What is multithreading?
Answer: Multithreading means various threads that run in a system.

Question: How does multithreading take place on a computer with a single CPU?
Answer: 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.

Question: How to create multithread in a program?
Answer: 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.

Question: Can Java object be locked down for exclusive use by a given thread?
Answer: 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

Question: Can each Java object keep track of all the threads that want to exclusively access to it?
Answer: Yes

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

Question: What invokes a thread's run() method?
Answer: 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.

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

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

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

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

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

Question: What is the Vector class?
Answer: 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.

Question: If a method is declared as protected, where may the method be accessed?
Answer: 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.

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

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

Question: What is the difference between yielding and sleeping?
Answer: 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.

Question: Is sizeof a keyword?
Answer: The sizeof operator is not a keyword.

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

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

Question: What is the difference between preemptive scheduling and time slicing?
Answer: 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.

Question: Name Component subclasses that support painting.
Answer: The Canvas, Frame, Panel, and Applet classes support painting.

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

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

Question: . Can an anonymous class be declared as implementing an interface and extending a class?
Answer: An anonymous class may implement an interface or extend a superclass, but may not be declared to do both.

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

Question: Which class is the superclass for every class.
Answer: Object.

Question: What is the difference between the Boolean & operator and the && operator?
Answer: 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.

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

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

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

Question: What is the Properties class?
Answer: 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.

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

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

Question: What is the purpose of the finally clause of a try-catch-finally statement?
Answer: The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

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

Question: What must a class do to implement an interface?
Answer: It must provide all of the methods in the interface and identify the interface in its implements clause.

Question: What is an abstract method?
Answer: An abstract method is a method whose implementation is deferred to a subclass. Or, a method that has no implementation (an interface of a method).

Question: What is a static method?
Answer: 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.

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

Question: What is the difference between a static and a non-static inner class?
Answer: 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.

Question: What is an object's lock and which object's have locks?
Answer: 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.

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

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

Question: What do heavy weight components mean?
Answer: 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.

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

Question: What are peerless components?
Answer: The peerless components are called light weight components.

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

Question: What happens when a thread cannot acquire a lock on an object?
Answer: 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.

Question: What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?
Answer: The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

Question: What classes of exceptions may be caught by a catch clause?
Answer: A catch clause can catch any exception that may be assigned to the Throwable type. This includes the Error and Exception types.

Question: What is the difference between throw and throws keywords?
Answer: 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 mehtod, 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.

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

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

Question: Does a class inherit the constructors of its superclass?
Answer: A class does not inherit constructors from any of its superclasses.

Question: Name primitive Java types.
Answer: The primitive types are byte, char, short, int, long, float, double, and boolean.

Question: Which class should you use to obtain design information about an object?
Answer: The Class class is used to obtain information about an object's design.

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

Question: How are the elements of a GridBagLayout organized?
Answer: 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.

Question: What advantage do Java's layout managers provide over traditional windowing systems?
Answer: 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.

Question: What are the problems faced by Java programmers who don't use layout managers?
Answer: 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.

Question: What is the difference between static and non-static variables?
Answer: 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.

Question: What is the difference between the paint() and repaint() methods?
Answer: 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.

Question: What is the purpose of the File class?
Answer: The File class is used to create objects that provide access to the files and directories of a local file system.

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

Question: What restrictions are placed on method overriding?
Answer: 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.

Question: What is casting?
Answer: 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.

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

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

Question: How are this() and super() used with constructors?
Answer: this() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

Question: How is it possible for two String objects with identical values not to be equal under the == operator?
Answer: 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.

Question: What an I/O filter?
Answer: 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.

Question: What is the Set interface?
Answer: The Set interface provides methods for accessing the elements of a finite mathematical set. Sets do not allow duplicate elements.

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

Question: What is the purpose of the enableEvents() method?
Answer: 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.

Question: What is the difference between the File and RandomAccessFile classes?
Answer: 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.

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

Question: What is the ResourceBundle class?
Answer: 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.

Question: What is the difference between a Scrollbar and a ScrollPane?
Answer: 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.

Question: What is a Java package and how is it used?
Answer: 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.

Question: What are the Object and Class classes used for?
Answer: 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.

Question: What is Serialization and deserialization?
Answer: Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

Question: what is tunnelling?
Answer: 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.

Question: Does the code in finally block get executed if there is an exception and a return statement in a catch block?
Answer: 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.

Question: How you restrict a user to cut and paste from the html page?
Answer: Using javaScript to lock keyboard keys. It is one of solutions.

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

Question: What is a Container in a GUI?
Answer: 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.

Question: How the object oriented approach helps us keep complexity of software development under control?
Answer:

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 allows constraints to be imposed on inheriting classes while still allowing the flexibility of overriding and overloading.

Question: What is polymorphism?
Answer: Polymorphism allows methods 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 yet unconceived classes.

Question: What is design by contract?
Answer: 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.

Question: What are use cases?
Answer: 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.

Question: What is the difference between interface and abstract class?
Answer:

- 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.

Question: What is an Iterator interface?

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

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

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

Question: Which method of the Component class is used to set the position and size of a component?

Answer: setBounds()

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

Answer: 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.

Question: What is the difference between yielding and sleeping?

Answer: 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.

Question: Which java.util classes and interfaces support event handling?

Answer: The EventObject class and the EventListener interface support event processing.

Question: Is sizeof a keyword?

Answer: The sizeof operator is not a keyword.

Question: What are wrapped classes?

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

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

Answer: 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

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

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

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

Answer: 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.

Question: What is the immediate superclass of the Applet class?

Answer: Panel

Question: What is the difference between preemptive scheduling and time slicing?

Answer: 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.

Question: Name three Component subclasses that support painting.

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

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

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

Question: What is the immediate superclass of the Dialog class?

Answer: Window

Question: What is clipping?

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

Question: What is a native method?

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

Question: Can a for statement loop indefinitely?

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

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

Answer: 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

Question: When a thread blocks on I/O, what state does it enter?

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

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

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

Question: What is the catch or declare rule for method declarations?

Answer: 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.

Question: What is the difference between a MenuItem and a CheckboxMenuItem?

Answer: The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.

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

Answer: 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.

Question: What class is the top of the AWT event hierarchy?

Answer: The java.awt.AWTEvent class is the highest-level class in the AWT event-class hierarchy.

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

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

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

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

Question: What is the range of the short type?

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

Question: What is the range of the char type?

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

Question: In which package are most of the AWT events that support the event-delegation model defined?

Answer: 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.

Question: What is the immediate superclass of Menu?

Answer: MenuItem

Question: What is the purpose of finalization?

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

Question: Which class is the immediate superclass of the MenuComponent class.

Answer: Object

Question: What invokes a thread's run() method?

Answer: 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.

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

Answer: 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.

Question: Name three subclasses of the Component class.

Answer: Box.Filler, Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, or TextComponent

Question: What is the GregorianCalendar class?

Answer: The GregorianCalendar provides support for traditional Western calendars.

Question: Which Container method is used to cause a container to be laid out and redisplayed?

Answer: validate()

Question: What is the purpose of the Runtime class?

Answer: The purpose of the Runtime class is to provide access to the Java runtime system.

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

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

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

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

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

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

Question: Which Java operator is right associative?

Answer: The = operator is right associative.

Question: What is the Locale class?

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

Question: Can a double value be cast to a byte?

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

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

Answer: 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.

Question: What must a class do to implement an interface?

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

Question: What method is invoked to cause an object to begin executing as a separate thread?

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

Question: Name two subclasses of the TextComponent class.

Answer: TextField and TextArea

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

Answer: 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.

Question: Which containers may have a MenuBar?

Answer: Frame

Question: How are commas used in the intialization and iterationparts of a for statement?

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

Question: What is the purpose of the wait(), notify(), and notifyAll() methods?

Answer: 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.

Question: What is an abstract method?

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

Question: How are Java source code files named?

Answer: 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.

Question: What is the relationship between the Canvas class and the Graphics class?

Answer: A Canvas object provides access to a Graphics object via its paint() method.

Question: What are the high-level thread states?

Answer: The high-level thread states are ready, running, waiting, and dead.

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

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

Question: Can a Byte object be cast to a double value?

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

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

Answer: 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.

Question: What is the difference between the String and StringBuffer classes?

Answer: String objects are constants. StringBuffer objects are not.

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

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

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

Answer: 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.

Question: What is the Dictionary class?

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

Question: How are the elements of a BorderLayout organized?

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

Question: What is the % operator?

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

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

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

Question: What is the difference between a Window and a Frame?

Answer: The Frame class extends Window to define a main application window that can have a menu bar.

Question: Which class is extended by all other classes?

Answer: The Object class is extended by all other classes.

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

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

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

Answer: It is written x ? y : z.

Question: What is the difference between the Font and FontMetrics classes?

Answer: The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object.

Question: How is rounding performed under integer division?

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

Question: What happens when a thread cannot acquire a lock on an object?

Answer: 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.

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

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

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

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

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

Answer: 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.

Question: What is the SimpleTimeZone class?

Answer: The SimpleTimeZone class provides support for a Gregorian calendar.

Question: What is the Map interface?

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

Question: Does a class inherit the constructors of its superclass?

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

Question: For which statements does it make sense to use a label?

Answer: The only statements for which it makes sense to use a label are those statements that can enclose a break or continue statement.

Question: What is the purpose of the System class?

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

Question: Which TextComponent method is used to set a TextComponent to the read-only state?

Answer: setEditable()

Question: How are the elements of a CardLayout organized?

Answer: The elements of a CardLayout are stacked, one on top of the other, like a deck of cards.

Question: Is &&= a valid Java operator?

Answer: No, it is not.

Question: Name the eight primitive Java types.

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

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

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

Question: What is the relationship between clipping and repainting?

Answer: When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.

Question: Is "abc" a primitive value?

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

Question: What is the relationship between an event-listener interface and an event-adapter class?

Answer: 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.

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

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

Question: What modifiers may be used with an interface declaration?

Answer: An interface may be declared as public or abstract.

Question: Is a class a subclass of itself?

Answer: A class is a subclass of itself.

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

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

Question: What event results from the clicking of a button?

Answer: The ActionEvent event is generated as the result of the clicking of a button.

Question: How can a GUI component handle its own events?

Answer: A component can handle its own events by implementing the required event-listener interface and adding itself as its own event listener.

Question: What is the difference between a while statement and a dostatement?

Answer: 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.

Question: How are the elements of a GridBagLayout organized?

Answer: 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.

Question: What advantage do Java's layout managers provide over traditional windowing systems?

Answer: 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 accomodate platform-specific differences among windowing systems.

Question: What is the Collection interface?

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

Question: What modifiers can be used with a local inner class?

Answer: A local inner class may be final or abstract.

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

Answer: 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.

Question: What is the difference between the paint() and repaint() methods?

Answer: 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.

Question: What is the purpose of the File class?

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

Question: Can an exception be rethrown?

Answer: Yes, an exception can be rethrown.

Question: Which Math method is used to calculate the absolute value of a number?

Answer: The abs() method is used to calculate absolute values.

Question: How does multithreading take place on a computer with a single CPU?

Answer: 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.

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

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

Question: When is the finally clause of a try-catch-finally statement executed?

Answer: 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.

Question: Which class is the immediate superclass of the Container class?

Answer: Component

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

Answer: 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.

Question: How can the Checkbox class be used to create a radio button?

Answer: By associating Checkbox objects with a CheckboxGroup.

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

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

Question: What restrictions are placed on method overloading?

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

Question: What happens when you invoke a thread's interrupt method while it is sleeping or waiting?

Answer: 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.

Question: What is casting?

Answer: 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.

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

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

Question: Name four Container classes.

Answer: Window, Frame, Dialog, File Dialog, Panel, Applet, or ScrollPane

Question: What is the difference between a Choice and a List?

Answer: 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.

Question: What class of exceptions are generated by the Java run-time system?

Answer: The Java runtime system generates Runtime Exception and Error exceptions.

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

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

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

Answer: 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.

Question: Under what conditions is an object's finalize() method invoked by the garbage collector?

Answer: The garbage collector invokes an object's finalize() method when it detects that the object has become unreachable.

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

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

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

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

Question: What is the difference between the JDK 1.02 event model and the event-delegation model introduced with JDK 1.1?

Answer: 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 unhand led events.

Question: How is it possible for two String objects with identical values not to be equal under the == operator?

Answer: 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.

Question: Why are the methods of the Math class static?

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

Question: What Checkbox method allows you to tell if a Checkbox is checked?

Answer: getState()

Question: What state is a thread in when it is executing?

Answer: An executing thread is in the running state.

Question: What are the legal operands of the instance of operator?

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

Question: How are the elements of a GridLayout organized?

Answer: The elements of a GridBad layout are of equal size and are laid out using the squares of a grid.

Question: What an I/O filter?

Answer: 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.

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

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

Question: What is the Set interface?

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

Question: What classes of exceptions may be thrown by a throw statement?

Answer: A throw statement may throw any expression that may be assigned to the Throwable type.

Question: What are E and PI?

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

Question: Are true and false keywords?

Answer: The values true and false are not keywords.

Question: What is a void return type?

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

Question: What is the purpose of the enableEvents() method?

Answer: 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.

Question: What is the difference between the File and RandomAccessFile classes?

Answer: 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.

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

Answer: The result is a String object.

Question: What is your platform's default character encoding?

Answer: 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..

Question: Which package is always imported by default?

Answer: The java.lang package is always imported by default.

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

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

Question: How are this and super used?

Answer: 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.

Question: What is the purpose of garbage collection?

Answer: 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.

Question: What is a compilation unit?

Answer: A compilation unit is a Java source code file.

Question: What interface is extended by AWT event listeners?

Answer: All AWT event listeners extend the java.util.EventListener interface.

Question: What restrictions are placed on method overriding?

Answer: 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.

Question: How can a dead thread be restarted?

Answer: A dead thread cannot be restarted.

Question: What happens if an exception is not caught?

Answer: 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.

Question: What is a layout manager?

Answer: A layout manager is an object that is used to organize components in a container.

Question: Which arithmetic operations can result in the throwing of an ArithmeticException?

Answer: Integer / and % can result in the throwing of an ArithmeticException.

Question: What are three ways in which a thread can enter the waiting state?

Answer: 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.

Question: Can an abstract class be final?

Answer: An abstract class may not be declared as final.

Question: What is the ResourceBundle class?

Answer: 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.

Question: 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?

Answer: The exception propagates up to the next higher level try-catch statement (if any) or results in the program's termination.

Question: What is numeric promotion?

Answer: 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.

Question: What is the difference between a Scrollbar and a ScrollPane?

Answer: 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.

Question: What is the difference between a public and a non-public class?

Answer: A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

Question: To what value is a variable of the boolean type automatically initialized?

The default value of the boolean type is false.

Question: Can try statements be nested?

Try statements may be tested.

Question: What is the difference between the prefix and postfix forms of the ++ operator?

The prefix form performs the increment operation and returns the value ofthe increment operation. The postfix form returns the current value all of the expression and then performs the increment operation on that value.

Question: What is the purpose of a statement block?

A statement block is used to organize a sequence of statements as a single statement group.

Question: 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.

Question: What modifiers may be used with a top-level class?

A top-level class may be public, abstract, or final.

Question: 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.

Question: 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.

Question: Can an unreachable object become reachable again?

Answer: 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.

Question: When is an object subject to garbage collection?

Answer: An object is subject to garbage collection when it becomes unreachable to the program in which it is used.

Question: What method must be implemented by all threads?

Answer: All tasks must implement the run() method, whether they are a subclass ofThread or implement the Runnable interface.

Question: What methods are used to get and set the text label displayed by a Button object?

Answer: getLabel() and setLabel()

Question: Which Component subclass is used for drawing and painting?

Answer: Canvas

Question: What are synchronized methods and synchronized statements?

Answer: 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.

Question: What are the two basic ways in which classes that can be run as threads may be defined?

Answer: A thread class may be declared as a subclass of Thread, or it may implement the Runnable interface.

Question: What are the problems faced by Java programmers who don't use layout managers?

Answer: Without layout managers, Java programmers are faced with determining how their GUI will be displayed across multiple windowing systems and finding a common sizingand positioning that will work within the constraints imposed by each windowing system.

Question: What is the difference between an if statement and a switch statement?

Answer: 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.

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

Answer: The result is a String object.

Question: What is the List interface?

Answer: The List interface provides support for ordered collections of objects.

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

Answer: We need

- public: The method can be accessed outside the class / package
- static: You need not have an instance of the class to access the method
- void: Your application need not return a value, as the JVM launcher would return the value when it exits
- main(): This is the entry point for the application

If the main() was not static, you would require an instance of the class in order to execute the method.
If this is the case, what would create the instance of the class? What if your class did not have a public constructor?

Question: What is the difference between an Interface and an Abstract class

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

Question: Explain serialization

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

Question: What are the rules of serialization

Answer: Rules:

1. Static fileds are not serialized because they are not part of any one particular object
2. Fileds from the base class are handled only if hose are serializable
3. Transient fileds are not serialized

Question: What is difference between error and exception

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

Question: What do you mean by object oreiented programming

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

Question: What are 4 pillars of object oreinted programming

Answer:

1. Abstraction
It means hiding the details and only exposing the essentioal parts

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

3. Inheritance
Inheritance means the child class inherits the non private properties of the parent class

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

Question: Difference between procedural and object oreinted language

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

Question: What is the difference between constructor and method

Answer: Constructor will be automatically invoked when an object is created whereas method has to be called explicitly.

Question: What is the difference between parameters and arguments

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

Question: What is reflection in java

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

Question: What is a cloneable interface and how many methods does it contain

Answer: It is not having any method because it is a TAGGED or MARKER interface

Question: What's the difference between a queue and a stack

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

Question: Can you make an instance of abstract class

Answer: No you cannot create an instance of abstract class

Question: What are parsers

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

Question: Difference between SAX and DOM parser

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

Question: What is the difference between Java Bean and Java Class

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

Question: What are null or Marker interfaces in Java

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

Question: Does java Support multiple inheritance

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

Question: What are virtual function

Answer: In OOP when a derived class inherits from a base class, an object of the derived class may be referred to (or cast) as either being the base class type or the derived class type. If there are base class functions overridden by the derived class, a problem then arises when a derived object has been cast as the base class type. When a derived object is referred to as being of the base's type, the desired function call behavior is ambiguous.

The distinction between virtual and not virtual is provided to solve this issue. If the function in question is designated "virtual" then the derived class's function would be called (if it exists). If it is not virtual, the base class's function would be called.

Question: Does java support virtual functions

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

Question: Describe what happens when an object is created in Java

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

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

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

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

4. Before the body of the constructor is executed, all instance variable initializers and initialization blocks
are executed. Then the body of the constructor is executed. Thus, the constructor for the base class
completes first and constructor for the most derived class completes last.

Question: What is the purpose of System Class

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

Question: What is *instanceOf* operator used for

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

Question: Why we should not have instance variable in an interface

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

Question: What is JVM

Answer: When we install a java package. It contains 2 things
* The Java Runtime Environment (JRE)
* The Java Development Kit (JDK)

The JRE provides runtime support for Java applications. The JDK provides the Java compiler and other development tools. The JDK includes the JRE.

Both the JRE and the JDK include a Java Virtual Machine (JVM). This is the application that executes a Java program. A Java program requires a JVM to run on a particular platform

Question: Can abstract class be final

Answer: No, abstract class cannot be final

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

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

Question: What is a singleton class

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

Question: Can an abstract class have final method

Answer: Yes

Question: Can a final class have an abstract method

Answer: No, the compiler will give an error

Question: What is the difference between Authentication and Authorization

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

Question: What are null or Marker interfaces in Java

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

Question: Does java Support multiple inheritance

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

Question: What are virtual function

Answer: In OOP when a derived class inherits from a base class, an object of the derived class may be referred to (or cast) as either being the base class type or the derived class type. If there are base class functions overridden by the derived class, a problem then arises when a derived object has been cast as the base class type. When a derived object is referred to as being of the base's type, the desired function call behavior is ambiguous.

The distinction between virtual and not virtual is provided to solve this issue. If the function in question is designated "virtual" then the derived class's function would be called (if it exists). If it is not virtual, the base class's function would be called.

Question: Does java support virtual functions

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

Question: Describe what happens when an object is created in Java

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

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

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

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

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

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

Question: What is the purpose of System Class

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

Question: What is *instanceOf* operator used for

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

Question: Why we should not have instance variable in an interface

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

Question: What is JVM

Answer: When we install a java package. It contains 2 things
* The Java Runtime Environment (JRE)
* The Java Development Kit (JDK)

The JRE provides runtime support for Java applications. The JDK provides the Java compiler and other development tools. The JDK includes the JRE.

Both the JRE and the JDK include a Java Virtual Machine (JVM). This is the application that executes a Java program. A Java program requires a JVM to run on a particular platform

Question: Can abstract class be final

Answer: No, abstract class cannot be final

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

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

Question: What is a singleton class

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

Question: Can an abstract class have final method

Answer: Yes

Question: Can a final class have an abstract method

Answer: No, the compiler will give an error

Question: What is the difference between Authentication and Authorization

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

Question: How many number of non-public class definitions can a source file have A source file can contain unlimited number of non-public class definitions List primitive data types, there size and there range (min, max)

Answer:

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

Question: What types of values does boolean variables take It only takes values true and false. Which primitive datatypes are signed.

Answer: All except char and Boolean

Question: Is char type signed or unsigned

Answer: char type is integral but unsigned. It range is 0 to 2^7-1

Question: What forms an integral literal can be

Answer: decimal, octal and hexadecimal, hence example it can be 28, 034 and 0x1c respectively

Question: What is the default value of Boolean

Answer: False

Question: Why is the main method static

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

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

Answer: class variable is a static variable and does not belong to instance of class but rather shared across all the instances
member variable belongs to a particular instance of class and can be called from any method of the class
automatic or local variable is created on entry to a method and has only method scope

Question: When are static and non static variables of the class initialized

Answer: The static variables are initialized when the class is loadedNon static variables are initialized just before the constructor is called

Question: When are automatic variable initialized

Answer: Automatic variable have to be initialized explicitly

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

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

Question: What is garbage collection

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

Question: Does System.gc and Runtime.gc() guarantee garbage collection

Answer: No

Operators and assignment

Question: What are different types of operators in java

Answer: Uniary ++, --, +, -, |, ~, ()
Arithmetic *, /, %,+, -
Shift <<, >>, >>>
Comparison =, instanceof, = =,!=Bitwise &, ^, |Short Circuit &&, ||Ternary ?:Assignment =

Question: How does bitwise (~) operator work

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

Question: What is a modulo operator %

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

Question: Can shift operators be applied to float types.

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

Question: What happens to the bits that fall off after shifting

Answer: They are discarded

Question: What values of the bits are shifted in after the shift

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

Question: What are access modifiers

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

Question: Can protected or friendly features be accessed from different packages

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

Question: How can you access protected features from another package

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

Question: What are the rules for overriding

Answer:

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

Question: Explain modifier final

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

Question: Can you change the reference of the final object

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

Question: Can abstract modifier be applied to a variable

Answer: No it is applied only to class and methods

Question: Can abstract class be instantiated

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

Question: When does the compiler insist that the class must be abstract

Answer:

If one or more methods of the class are abstract.
If class inherits one or more abstract methods from the parent abstract class and no implementation is provided for that method
If class implements an interface and provides no implementation for those methods

Question: How is abstract class different from final class

Answer: Abstract class must be subclassed and final class cannot be subclassed

Question: Where can static modifiers be used

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

Question: When are the static variables loaded into the memory

Answer: During the class load time

Question: When are the non static variables loaded into the memory

Answer: They are loaded just before the constructor is called

Question: How can you reference static variables

Answer: Via reference to any instance of the class

Computer comp = new Computer ();
comp.harddisk where hardisk is a static variable
comp.compute() where compute is a method

Via the class name

Computer.harddisk
Computer.compute()

Question: Can static method use non static features of there class

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

Question: What is static initializer code

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

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

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

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

Question: What are transient variables

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

Question: What is synchronized modifier used for

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

Question: What are volatile variables

Answer: It indicates that these variables can be modified asynchronously

Question: What are wrapped classes

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

Question: What are the four general cases for Conversion and Casting

Answer:

Conversion of primitives
Casting of primitives
Conversion of object references
Casting of object references

Question: When can conversion happen

Answer:

It can happen during

Assignment
Method call
Arithmetic promotion

Question: What are the rules for primitive assignment and method call conversion

Answer:

A boolean can not be converted to any other type
A non Boolean can be converted to another non boolean type, if the conversion is widening conversion
A non Boolean cannot be converted to another non boolean type, if the conversion is narrowing conversion
See figure below for simplicity

Question: What are the rules for primitive arithmetic promotion conversion

Answer:

For Unary operators :

If operant is byte, short or a char it is converted to an int
If it is any other type it is not converted

For binary operands :

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

Question: What are the rules for casting primitive types

Answer:

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

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

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

Question: What are the rules for Object reference casting

Answer: Casting from Old types to Newtypes
Compile time rules

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

- When both Oldtype ad Newtype are arrays, both arrays must contain reference types (not primitive), and it must be legal to cast an element of Oldtype to an element of Newtype
- You can always cast between an interface and a non-final object

Runtime rules

- If Newtype is a class. The class of the expression being converted must be Newtype or must inherit from Newtype
- If NewType is an interface, the class of the expression being converted must implement Newtype

Question: What is the difference between while and do while loop

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

Question: When do you use continue and when do you use break statements

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

Question: What is the base class from which all exceptions are subclasses

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

Question: How do you intercept and thereby control exceptions

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

Question: When do we say an exception is handled

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

Question: When do we say an exception is not handled

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

Question: In what sequence does the finally block gets executed

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

Question: What can prevent the execution of the code in finally block

Answer:

- The death of thread
- Use of system.exit()
- Turning off the power to CPU
- An exception arising in the finally block itself

What are the rules for catching multiple exceptions
A more specific catch block must precede a more general one in the source, else it gives compilation error
Only one catch block, that is first applicable one, will be executed

Question: What does throws statement declaration in a method indicate

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

Question: What are checked exception

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

Question: What are runtime exceptions

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

Question: What is difference between Exception and errors

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

Question: How will you handle the checked exceptions

Answer: You can provide a try/catch block to handle it. OR Make sure method declaration includes a throws clause that informs the calling method an exception might be thrown from this particular method. When you extend a class and override a method, can this new method throw exceptions other than those that were declared by the original method. No it cannot throw, except for the subclasses of those exceptions.

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

Answer: Yes it is perfectly legal

Question: Explain the user defined Exceptions?

Answer: 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
}
```

Objects and Classes

Question: What's the difference between constructors and other methods

Answer: 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.

Question: What is the difference between Overloading and Overriding

Answer: Overloading : Reusing the same method name with different arguments and perhaps a different return type is called as overloading
Overriding : Using the same method name with identical arguments and return type is know as overriding

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

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

Question: Can overriding methods have different return types

Answer: No they cannot have different return types

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

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

Question: Can constructors be overloaded

Answer: Yes constructors can be overloaded

Question: What happens when a constructor of the subclass is called

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

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

Answer: It should always be the first statement in the constructor

Question: What is an inner class

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

Question: How will you reference the inner class

Answer: To reference it you will have to use OuterClass$InnerClass

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

Answer: Yes they can access the members of the outer class

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

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

Question: Can inner classes be static

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

Question: Can an inner class be defined inside a method

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

Question: What is an anonymous class

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

Question: What are the rules of anonymous class

Answer: The class is instantiated and declared in the same place The declaration and instantiation takes the form new Xxxx () {// body}
Where Xxxx is an interface name. An anonymous class cannot have a constructor. Since you do not specify a name for the class, you cannot use that name to specify a constructor

Question: Where does java thread support reside

Answer: It resides in three places

The java.lang.Thread class (Most of the support resides here)
The java.lang.Object class
The java language and virtual machine

Question: What is the difference between Thread and a Process

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

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

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

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

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

Question: When the thread gets to execute, what does it execute

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

Question: How many methods are declared in the interface runnable

Answer: The runnable method declares only one method :
public void run();

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

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

Question: What happens when the run() method returns

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

Question: What are the different states of the thread

Answer: They are as follows:

Running: The state that all thread aspire to be
Various waiting states : Waiting, Sleeping, Suspended and Bloacked

Ready : Waiting only for the CPU
Dead : All done

Question: What is Thread priority

Answer: Every thread has a priority, the higher priorit thread gets preference over the lower priority thread by the thread scheduler

Question: What is the range of priority integer

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

Question: What is the default priority of the thread

Answer: The default priority is 5

Question: What happens when you call Thread.yield()

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

Question: What is the advantage of yielding

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

Question: What happens when you call Thread.sleep()

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

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

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

Question: What do you mean by thread blocking

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

Question: What threading related methods are there in object class

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

Question: What is preemptive scheduling

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

Question: What is non-preemptive or Time sliced or round robin scheduling

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

Question: What are the two ways of synchronizing the code

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

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

Question: What happens when the wait() method is called

Answer: The calling thread gives up CPU
The calling thread gives up the lock
The calling thread goes into the monitor's waiting pool

Question: What happens when the notify() method is called

Answer: One thread gets moved out of monitors waiting pool and into the ready state
The thread that was notified ust reacquire the monitors locl before it can proceed

Question: Using notify () method how you can specify which thread should be notified

Answer: You cannot specify which thread is to be notified, hence it is always better to call notifyAll() method

Question: What is the ultimate ancestor of all java classes

Answer: Object class is the ancestor of all the java classes

Question: What are important methods of Object class

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

Question: What is the difference between "= =" and "equals()"

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

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

Answer: 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

Question: Give example of a final class

Answer: Math class is final class and hence cannot be extended

Question: What is the difference between String and StringBuffer

Answer: String is an immutable class, i.e you cannot change the values of that class

Example:

String str = “java”; // address in memory say 12345
And now if you assign a new value to the variable str then
str = “core java”; then the value of the variable at address 12345 will not change but a new memory is allocated for this variable say 54321
So in the memory address 12345 will have value “java”
And the memory address 54321 will have value “core java” and the variable str will now be pointing to address 54321 in memory

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

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

Answer: It will return false as you cannot compare String with StringBuffer

Question: What is Collection API

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

Question: What are different types of collections

Answer: A collection has no special order and does not reject duplicates
A list is ordered and does not reject duplicates
A set has no special order but rejects duplicates
A map supports searching on a key field, values of which must be unique

Question: Tell me something about Arrays

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

Question: Difference between ArrayList and Vector

Answer: Vector methods are synchronized while ArrayList methods are not

Question: 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

Question: Difference between Hashtable and HashMap

Answer: Hashtable does not store null value, while HashMap does
Hashtable is synchronized, while HashMap is not

Reference: www.roseindia.com

www.google.com

----------BEST OF LUCK----------