1 :: What is a transient variable?

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

2 :: Which containers use a border layout as their default layout?

The window, Frame and Dialog classes use a border layout as their default layout in Java Programming.

3 :: How are Observer and Observable used in Java Programming?

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

4 :: What is synchronization and why is it important in Java Programming?

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.

5 :: What are synchronized methods and synchronized statements in Java Programming?

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.

6 :: What are three ways in which a thread can enter the waiting state in Java Programming?

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.

7 :: Can a lock be acquired on a class in Java Programming?

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

8 :: What's new with the stop(), suspend() and resume() methods in JDK 1.2?

The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

9 :: What is the preferred size of a component in Java Programming?

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

10 :: What method is used to specify a container's layout in Java Programming?

The setLayout() method is used to specify a container's layout in Java Programming.

11 :: Which containers use a FlowLayout as their default layout in Java Programming?

The Panel and Applet classes use the FlowLayout as their default layout.

12 :: What state does a thread enter when it terminates its processing in Java Programming?

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

13 :: What is the Collections API in Java Programming?

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

14 :: What is the List interface in Java Programming?

The List interface provides support for ordered collections of objects.

15 :: How does Java handle integer overflows and underflows?

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

16 :: What is the Vector class in Java Programming?

The Vector class in Java Programming provides the capability to implement a growable array of objects

17 :: What modifiers may be used with an inner class that is a member of an outer class in Java Programming?

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

18 :: If a method is declared as protected, where may the method be accessed in Java Programming?

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.

19 :: What is an Iterator interface in Java Programming?

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

20 :: How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters in Java Programming?

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.

21 :: What is the difference between yielding and sleeping in Java Programming?

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.

22 :: Is sizeof a keyword in Java Programming?

The sizeof operator is not a keyword in Java Programming.

23 :: What are wrapped classes in Java Programming?

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

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

No, it doesn’t. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection

25 :: What is the difference between preemptive scheduling and time slicing in Java Programming?

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.

26 :: Name Component subclasses that support painting in Java Programming?

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

27 :: What is a native method in Java Programming?

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

28 :: How can you write a loop indefinitely in Java Programming?

for(;;)–for loop; while(true)–always true, etc.

29 :: Can an anonymous class be declared as implementing an interface and extending a class in Java Programming?

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

30 :: What is the purpose of finalization in Java Programming?

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

31 :: Which class is the superclass for every class in Java Programming?

Object class is the superclass for every class in Java Programming

32 :: What invokes a thread's run() method in Java Programming?

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.

33 :: What is the difference between the Boolean & operator and the && operator in Java Programming?

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.

34 :: What is the GregorianCalendar class in Java Programming?

The GregorianCalendar provides support for traditional Western calendars in Java Programming.

35 :: What is the SimpleTimeZone class in Java Programming?

The SimpleTimeZone class provides support for a Gregorian calendar in Java Programming.

36 :: Which Container method is used to cause a container to be laid out and redisplayed in Java Programming?

validate();
Container method is used to cause a container to be laid out and redisplayed in Java Programming

37 :: What is the Properties class in Java Programming?

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.

38 :: What is the purpose of the Runtime class in Java Programming?

The purpose of the Runtime class is to provide access to the Java runtime system in Java Programming.

39 :: What is the purpose of the System class in Java Programming?

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

40 :: What is the purpose of the finally clause of a try-catch-finally statement in Java Programming?

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

41 :: What is the Locale class in Java Programming?

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

42 :: What must a class do to implement an interface in Java Programming?

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

43 :: What is the purpose of the wait(), notify(), and notifyAll() methods in Java Programming?

The wait(),notify(), and notifyAll() methods are used to provide an efficient way for threads to communicate each other in Java Programming.

44 :: What is an abstract method in Java Programming?

An abstract method is a method whose implementation is deferred to a subclass in Java Programming.

45 :: What are the high-level thread states in Java Programming?

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

46 :: What is the difference between a static and a non-static inner class in Java Programming?

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.

47 :: What is an object's lock and which object's have locks in Java Programming?

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.

48 :: When can an object reference be cast to an interface reference in Java Programming?

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

49 :: What is the difference between a Window and a Frame in Java Programming?

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

50 :: What do heavy weight components mean in Java Programming?

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 heavyweight.

51 :: Which package has light weight components in Java Programming?

javax.Swing package. All components in Swing, except JApplet, JDialog, JFrame and JWindow are lightweight components in Java Programming.

52 :: What are peerless components in Java Programming?

The peerless components are called light weight components.

53 :: What is the difference between the Font and FontMetrics classes in Java Programming?

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

54 :: What happens when a thread cannot acquire a lock on an object in Java Programming?

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.

55 :: What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy in Java Programming?

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

56 :: What classes of exceptions may be caught by a catch clause in Java Programming?

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

57 :: If a class is declared without any access modifiers, where may the class be accessed in Java Programming?

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.

58 :: What is the Map interface in Java Programming?

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

59 :: Does a class inherit the constructors of its superclass in Java Programming?

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

60 :: Name primitive Java types?

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

61 :: Which class should you use to obtain design information about an object in Java Programming?

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

62 :: How can a GUI component handle its own events in Java Programming?

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

63 :: How are the elements of a GridBagLayout organized in Java Programming?

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.

64 :: What advantage do Java’s layout managers provide over traditional windowing systems?

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

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

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

66 :: What is the difference between static and non-static variables in Java Programming?

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.

67 :: What is the difference between the paint() and repaint() methods in Java Programming?

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.

68 :: What is the purpose of the File class in Java Programming?

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

69 :: How does multithreading take place on a computer with a single CPU in Java Programming?

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.

70 :: What restrictions are placed on method overloading in Java Programming?

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

71 :: What restrictions are placed on method overriding in Java Programming?

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.

72 :: What is casting in Java Programming?

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.

73 :: Name Container classes in Java Programming?

Container classes in Java:

Window,
Frame,
Dialog,
FileDialog,
Panel,
Applet,
or ScrollPane

74 :: What class allows you to read objects directly from a stream in Java Programming?

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

75 :: How are this() and super() used with constructors in Java Programming?

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

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

The == operator compares two objects to determine if they are the same object in memory. It is possible for two String objects to have the same value, but located indifferent areas of memory.

77 :: What an I/O filter in Java Programming?

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.

78 :: What is the Set interface in Java Programming?

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

79 :: What is the purpose of the enableEvents() method in Java Programming?

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.

80 :: Is Java a super set of JavaScript?

No. They are completely different. Some syntax may be similar.