

## Interview Questions in Core Java

### **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.

### **3.Why do threads block on I/O?**

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

### **4. How are Observer and Observable used?**

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

### **5. What is synchronization and why is it important?**

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

### **6. Can a lock be acquired on a class?**

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

### **7. 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.

### **8. Is null a keyword?**

The null value is not a keyword.

### **9. What is the preferred size of a component?**

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

### **10. What method is used to specify a container's layout?**

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

### **11. Which containers use a FlowLayout as their default layout?**

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

### **12. What state does a thread enter when it terminates its processing?**

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

### **13. What is the Collections API?**

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

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

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

### **15. What is the List interface?**

The List interface provides support for ordered collections of objects.

**16. 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.

**17. What is the Vector class?**

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

**18. 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.

**19. What is an Iterator interface?**

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

**20. What is the difference between the >> and >>> operators?**

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

**21. Which method of the Component class is used to set the position and size of a component?**

setBounds()

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

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

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

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

**24. Which java.util classes and interfaces support event handling?**

The EventObject class and the EventListener interface support event processing.

**25. Is sizeof a keyword?**

The sizeof operator is not a keyword.

**26. What are wrapped classes?**

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

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

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

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

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

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

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

**30. What is the immediate superclass of the Applet class?**

Panel

**31. What is the difference between preemptive scheduling and time slicing?**

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

**32. Name three Component subclasses that support painting.**

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

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

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

**34. What is the immediate superclass of the Dialog class?**

Window

**35. What is clipping?**

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

**36. What is a native method?**

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

**37. Can a for statement loop indefinitely?**

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

```
for(;;) ;
```

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

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

**39. When a thread blocks on I/O, what state does it enter?**

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

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

The default value of an String type is null.

**41. What is the catch or declare rule for method declarations?**

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

**42. What is the difference between a MenuItem and a CheckboxMenuItem?**

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

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

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

**44. What class is the top of the AWT event hierarchy?**

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

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

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

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

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