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.

23What 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. Associatity 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.

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

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

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

50. What is the immediate superclass of Menu?
MenuItem

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

52. Which class is the immediate superclass of the MenuComponent class.
Object

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

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

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

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

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

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

59. How many times may an object's finalize() method be invoked by the garbage collector?
An object's finalize() method may only be invoked once by the garbage collector.

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

61. What is the argument type of a program's main() method?
A program's main() method takes an argument of the String[] type.

62. Which Java operator is right associative?
The = operator is right associative.

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

64. Can a double value be cast to a byte?
Yes, a double value can be cast to a byte.

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

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

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

68. Name two subclasses of the TextComponent class.
TextField and TextArea

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

70. Which containers may have a MenuBar?
Frame

71. How are commas used in the initialization and iteration parts of a for statement?
Commas are used to separate multiple statements within the initialization and iteration parts of a for statement.

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

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

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

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

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

77. What value does read() return when it has reached the end of a file?
The read() method returns -1 when it has reached the end of a file.

78. Can a Byte object be cast to a double value?
No, an object cannot be cast to a primitive value.

79. What is the difference between a static and a non-static inner class?
A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

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

81. If a variable is declared as private, where may the variable be accessed?
A private variable may only be accessed within the class in which it is declared.

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

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

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

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

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

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

88. Which class is extended by all other classes?
The Object class is extended by all other classes.

89. Can an object be garbage collected while it is still reachable?
A reachable object cannot be garbage collected. Only unreachable objects may be garbage collected..

90. Is the ternary operator written x : y ? z or x ? y : z ?
It is written x ? y : z.

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

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

93. What happens when a thread cannot acquire a lock on an object?

If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

109. What is the relationship between an event-listener interface and an event-adapter class?

An event-listener interface defines the methods that must be implemented by an event handler for a particular kind of event. An event adapter provides a default implementation of an event-listener interface.

110. What restrictions are placed on the values of each case of a switch statement?
During compilation, the values of each case of a switch statement must evaluate to a value that can be promoted to an int value.

111. What modifiers may be used with an interface declaration?
An interface may be declared as public or abstract.

112. Is a class a subclass of itself?
A class is a subclass of itself.

113. What is the highest-level event class of the event-delegation model?
The java.util.EventObject class is the highest-level class in the event-delegation class hierarchy.

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

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

116. What is the difference between a while statement and a do statement?
A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

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

118. 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 accomodate platform-specific differences among windowing systems.

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

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

121. What is the difference between static and non-static variables?
A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

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

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

124. Can an exception be rethrown?
Yes, an exception can be rethrown.

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

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

127. When does the compiler supply a default constructor for a class?
The compiler supplies a default constructor for a class if no other constructors are provided.

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

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

130. If a method is declared as protected, where may the method be accessed?
A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

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

132. Which non-Unicode letter characters may be used as the first character of an identifier?
The non-Unicode letter characters $ and _ may appear as the first character of an identifier

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

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

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

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

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

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

139. What class of exceptions are generated by the Java run-time system?
The Java runtime system generates RuntimeException and Error exceptions.

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

141. What is the difference between a field variable and a local variable?
A field variable is a variable that is declared as a member of a class. A local variable is a variable that is declared local to a method.

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

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

144. What is the relationship between a method's throws clause and the exceptions that can be thrown during the method's execution?
A method's throws clause must declare any checked exceptions that are not caught within the body of the method.

145. What is the difference between the JDK 1.02 event model and the event-delegation model introduced with JDK 1.1?
The JDK 1.02 event model uses an event inheritance or bubbling approach. In this model, components are required to handle their own events. If they do not handle a particular event, the event is inherited by (or bubbled up to) the component's container. The container then either handles the event or it is bubbled up to its container and so on, until the highest-level container has been tried.

In the event-delegation model, specific objects are designated as event handlers for GUI components. These objects implement event-listener interfaces. The event-delegation model is more efficient than the event-inheritance model because it eliminates the processing required to support the bubbling of unhandled events.

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

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

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

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

150. What are the legal operands of the instanceof operator?
The left operand is an object reference or null value and the right operand is a class, interface, or array type.

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

152. What an I/O filter?
An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

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

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

155. What classes of exceptions may be thrown by a throw statement?
A throw statement may throw any expression that may be assigned to the Throwable type.

156. What are E and PI?
E is the base of the natural logarithm and PI is mathematical value pi.

157. Are true and false keywords?
The values true and false are not keywords.

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

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

160. What is the difference between the File and RandomAccessFile classes?
The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

161. What happens when you add a double value to a String?
The result is a String object.

162. What is your platform's default character encoding?
If you are running Java on English Windows platforms, it is probably Cp1252. If you are running Java on English Solaris platforms, it is most likely 8859_1..

163. Which package is always imported by default?
The java.lang package is always imported by default.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

185. 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.

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

187. 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.

188. 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.

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

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

191. What method must be implemented by all threads?
All tasks must implement the run() method, whether they are a subclass of Thread or implement the Runnable interface.

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

193. Which Component subclass is used for drawing and painting?
Canvas

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

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

196. 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.

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

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

Most frequent questions

0) Q: Java and C++
   A: Some of the similarities and differences are in the table:

Features	Java	C/C++
Pointer	No	Yes
Operator Overload	No	Yes
Typedef, Define, Preprocessors	No	Yes
Structures, Unions	No	Yes
Enums	No	Yes
Functions	No (only methods within classes)	Yes
Goto statement	No	Yes
Automatic Coercions	No(types should be converted explicitly)	Yes
Global Variables	No. Variable is part of a class	Yes
Templates	No	Yes
Private, Protected, Public Inheritance	No	Yes
Default parameters	No	Yes
Garbage Collection	Yes	No
Multi-thread support	Yes	No
Multiple Inheritance	Yes. Supports only interface inheritance and not implementation inheritance!	Yes
Exception Handling	Yes. try/catch must be defined if the function declares that it may throw an exception.	Yes. You may not include the try/catch even if the function throws an exception.
Function Overload	Yes	Yes
Internationalization	Yes	Yes
Include of other Objects	#import	#include
Comments	"//","/* */,/** */	"//","/* */"

1) Q: What is the purpose of the toolkit in the Abstract Window Toolkit (AWT)? How does AWT work ?
   A: The AWT toolkit is an interface between the abstract window layer and a specific windowing implementation.

 2) Q: What is layout manager ? How does it work ?

A: A layout manager is an object that positions and resizes the components in a Container according to some algorithm; for example, the FlowLayout layout manager lays out components from left to right until it runs out of room and then continues laying out components below that row.

3) Q: Advantages and disadvantages of layout manager ?

4) Q: Compare SWING components to standard AWT.
A: Swing is an extension of, and not a replacement for the AWT. There is some overlap between AWT and Swing (for example a Swing JButton component might be viewed as an improved functional replacement for an AWT Button component.) One of the advantages of Swing components is that because the components are not rendered on the screen by the operating system, the look and feel of a component does not change as the application or applet is executed on different platforms running under different operating systems. Furthermore, it is possible to cause Swing components to mimic the look and feel of a specific platform no matter what platform the program is running on. This is known as pluggable look and feel. Swing components support the JDK 1.1
Delegation Event Model. From an event handling viewpoint, Swing components operate the same as AWT components (except that Swing provides a number of new event types). Many Swing components don't have an AWT counterpart. A number of new and exciting components are included in the Swing library that don't exist in the AWT (tooltips, progress bars, trees, etc.)

5) Q: What is Java Beans ?
A: According to JavaSoft, "A Java Bean is a reusable software component that can be manipulated visually in a builder tool."

6) Q: What you know about Corba implementation in Java ?
A: Java 1.2 promises full CORBA IDL support.

7) Q: What do you know about networking support in Java ?
A: Java supports "low-level" and "high-level" classes. "Low-level" classes provide support for socket programming: Socket, DatagramSocket, and ServerSocket classes. "High-level" classes provide "Web programming": URL, URLEncoder, and URLConnection classes. Networking programming classes ease the programming of network applications, but do not substitute your knowledge of networking. Java networking like anything else in Java is platform-independent.

8) Q: What is it object serialization ?
A: Serialization is a way to convert objects (including complex data structures such as lists and trees) into a stream of bytes.

9) Q: How to make application thread-safe ?
A: You should use the word synchronized to mark the critical section of code. You may also use other methods of thread synchronization (see wait(), notify(), notifyAll() etc.)

10) Q: What is it reflection (introspection) ? Why is reflection possible in the Java language?
A: Reflection (introspection) is querying a class about its properties, and operating on methods and fields by the name for a given object instance. Reflection is possible in the Java language because of late binding.

11) Q: Why are Java ARchive (JAR) files important?
A: JAR files bundle .class files and optimize applet downloads.

Following answer may not be correct

12) Describe what happens when an object is created in Java

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.

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

The main reason you cannot create a button by
Button bt1= "abc";

is because "abc" is a literal string (something slightly different than a String object, by-the-way) and bt1 is a Button object. That simple. The only object in Java that can be assigned a literal String is java.lang.String. Important to not that you are NOT calling a java.lang.String constuctor when you type String s = "abc";

For example
                      String x = "abc";
                      String y = "abc";
refer to the same object. While
                      String x1 = new String("abc");
                      String x2 = new String("abc");
                      refer to two different objects.

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

Everything is an object in Java(Single root hierarchy as everything gets derived from java.lang.Object)

Java does not have all the complicated aspects of C++ (For ex: Pointers, templates, unions, operator overloading, structures etc..)

The Java language promoters initially said "No pointers!", but when many programmers questioned how you can work without pointers, the promoters began saying "Restricted pointers." You can make up your mind whether it's really a pointer or not. In any event, there's no pointer arithmetic.

There are no destructors in Java. (automatic garbage collection)

Java does not support conditional compile (#ifdef/#ifndef type).
Thread support is built into java but not in C++.

Java does not support default arguments. There's no scope resolution operator :: in Java. Java uses the dot for everything, but can get away with it since you can define elements only within a class. Even the method definitions must always occur within a class, so there is no need for scope
resolution there either.
There's no "goto " statement in Java.

Java doesn't provide multiple inheritance (MI), at least not in the same sense that C++ does.
Exception handling in Java is different because there are no destructors.

Java has method overloading, but no operator overloading. The String class does use the + and += operators to concatenate strings and String expressions use automatic type conversion, but that's a special built-in case.
Java is interpreted for the most part and hence platform independent.

15) What are interfaces?

Interfaces provide more sophisticated ways to organize and control the objects in your system.
The interface keyword takes the abstract concept one step further. You could think of it as a "pure" abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields, but The interface keyword takes the abstract concept one step further. You could think of it as a "pure" abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields, but

An interface says: "This is what all classes that implement this particular interface will look like." Thus, any code that uses a particular interface knows what methods might be called for that interface, and that's all. So the interface is used to establish a "protocol" between classes. (Some object-oriented programming languages have a keyword called protocolto do the same thing.)

15) How can you achieve Multiple Inheritance in Java?

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

16) What is the difference between StringBuffer and String class?

A string buffer implements a mutable sequence of characters. A string buffer is like a String, but can be modified. At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls.
The String class represents character strings. All string literals in Java programs, such as "abc" are constant and implemented as instances of this class; their values cannot be changed after they are created.

17) Describe, in general, how java's garbage collector works?

The Java runtime environment deletes objects when it determines that they are no longer being used. This process is known as garbage collection.
The Java runtime environment supports a garbage collector that periodically frees the memory used by objects that are no longer needed. The Java garbage collector is a mark-sweep garbage collector that scans Java's dynamic memory areas for objects, marking those that are referenced. After all possible paths to objects are investigated, those objects that are not marked (i.e. are not referenced) are known to be garbage and are collected.

18) What's the difference between == and equals method?
The equals method can be considered to perform a deep comparison of the value of an object, whereas the == operator performs a shallow comparison.
The equals() method compares the characters inside a string object. == operator compares two object references to check whether they refer to the same instances or not.

19) What are abstract classes, abstract methods?
Simply speaking a class or a method qualified with "abstract" keyword is an abstract class or abstract method.

You create an abstract class when you want to manipulate a set of classes through a common interface. All derived-class methods that match the signature of the base-class declaration will be called using the dynamic binding mechanism.
An abstract method is an incomplete method. It has only a declaration and no method body. Here is the syntax for an abstract method declaration: abstract void f();

20) How can you force all derived classes to implement a method present in the base class?
Creating and implementing an interface would be the best way for this situation. Just create an interface with empty methods which forces a programmer to implement all the methods present under it.
Another way of achieving this task is to declare a class as abstract with all its methods abstract.

21) What is the difference between an Applet and an Application?
1. Applets can be embedded in HTML pages and downloaded over the Internet whereas Applications have no special support in HTML for embedding or downloading.

2. Applets can only be executed inside a java compatible container, such as a browser or appletviewer whereas Applications are executed at command line by java.exe or   jview.exe.

3. Applets execute under strict security limitations that disallow certain operations(sandbox model security) whereas Applications have no inherent security restrictions.

4. Applets don't have the main() method as in applications. Instead they operate on an entirely different mechanism where they are initialized by init(),started by start(),stopped by stop() or destroyed by destroy().

22) Java says "write once, run anywhere". What are some ways this isn't quite true?
Any time you use system calls specific to one operating system and do not create alternative calls for another operating system, your program will not function correctly.

Solaris systems and Intel systems order the bits of an integer differently. (You may have heard of little endian vs. big endian)

If your code uses bit shifting, or other binary operators, they will not work on systems that have opposide endianism.

23) Describe java's security model.
Java's security model is one of the most interesting and unique aspects of the language. For the most part it's broken into two pieces: the user adjustable security manager that checks various API operations like file access, and the byte code verifier that asserts the validity of compiled byte code.

public abstract class SecurityManager java.lang.SecurityManager is an abstract class which different applications subclass to implement a particular security policy. It allows an application to determine whether or not a particular operation will generate a security exception.

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

The vector container class generalizes the concept of an ordinary C array. Like an array, a vector is an indexed data structure, with index values that range from 0 to one less than the number of elements contained in the structure. Also like an array, values are most commonly assigned to and extracted from the vector using the subscript operator. However, the vector differs from an array in the following important respects:

The size of the vector can change dynamically. New elements can be inserted on to the end of a vector, or into the middle. It is important to note, however, that while these abilities are provided, insertion into the middle of a vector is not as efficient as insertion into the middle of a list.

A vector has more "self-knowledge" than an ordinary array. In particular, a vector can be queried about its size, about the number of elements it can potentially hold (which may be different from its current size), and so on.
A vector can only hold references to objects and not primitive types.
Vector Implementaions are usually slower then array because of all the functionality that comes with them. As implemented in Java, vector is a thread-safe class and hence all methods are synchronous methods, which makes them considerably slow.

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

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

Type 3: JDBC-Net Pure Java Driver:
The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.

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

26) What does the keyword "synchronize" mean in java. When do you use it? What are the disadvantages of synchronization?
Synchronize is used when u want to make ur methods thread safe. The disadvantage of synchronise is it will end up in slowing down the program. Also if not handled properly it will end up in dead lock.

1. Only use (and minimize it's use)synchronization when writing multithreaded code as there is a speed (up to five to six time slower, depending on the execution time of the synchronized/non-synchronized method) cost associated with its use.
2. In case of syncronized method modifier, the byte code generated is the exact same as non-syncronized method. The only difference is that a flag called ACC_SYNCRONIZED property flag in method's method_info structure is set if the syncronized method modifier is present.
3. Also, syncronized keyword can make the code larger in size if used in the body of the method as bytecode for monitorenter/monitorexit is generated in addition to any exception handling.

27) What are native methods? How do you use them?
Native methods are methods that are defined as public static methods within a java class, but whose implementation is provided in another programming language such as C.

28) What is RMI?
RMI stands for Remote Method Invocation. Traditional approaches to executing code on other machines across a network have been confusing as well as tedious and error-prone to implement. The nicest way to think about this problem is that some object happens to live on another machine, and that you can send a

message to the remote object and get a result as if the object lived on your local machine. This simplification is exactly what Java Remote Method Invocation (RMI) allows you to do.

29) What is JDBC? Describe the steps needed to execute a SQL query using JDBC.
The JDBC is a pure Java API used to execute SQL statements. It provides a set of classes and interfaces that can be used by developers to write database applications.

The steps needed to execute a SQL query using JDBC:
1. Open a connection to the database.
2. Execute a SQL statement.
3. Process th results.
4. Close the connection to the database.

30) Access specifiers: "public", "protected", "private", nothing?
Public – any other class from any package can instantiate and execute the classes and methods
Protected – only subclasses and classes inside of the package can access the classes and methods
Private – the original class is the only class allowed to executed the methods.

31) What does the "final" keyword mean in front of a variable? A method? A class?
FINAL for a variable : value is constant
FINAL for a method : cannot be overridden
FINAL for a class : cannot be derived

32) Does Java have "goto"?
no
33) Why "bytecode"? Can you reverse-engineer the code from bytecode?

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

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

36) Does Java have destructors?
No garbage collector does the job working in the background

37) What does the "abstract" keyword mean in front of a method? A class?
Abstract keyword declares either a method or a class. If a method has a abstract keyword in front of it,it is called abstract method.Abstract method hs no body.It has only arguments and return type.Abstract methods act as placeholder methods that are implemented in the subclasses.

Abstract classes can't be instantiated.If a class is declared as abstract,no objects of that class can be created.If a class contains any abstract method it must be declared as abstract

38) Name four methods every Java class will have.
public String toString();
        public Object clone();
        public boolean equals();
        public int hashCode();

39) Given a text file, input.txt, provide the statement required to open
        this file with the appropriate I/O stream to be able to read and process this file.

40) Discuss the differences between creating a new class, extending a class and implementing an interface; and when each would be appropriate.

*Creating a new class is simply creating a class with no extensions and no
  implementations. The signature is as follows

  public class MyClass()
  {
  }

   *Extending a class is when you want to use the functionality of another class or
   classes. The extended class inherits all of the functionality of the previous class. An
   example of this when you create your own applet class and extend from
   java.applet.Applet. This gives you all of the functionality of the java.applet.Applet class.

   The signature would look like this

   public class MyClass extends MyBaseClass
   {
   }

   *Implementing an interface simply forces you to use the methods of the interface
   implemented. This gives you two advantages. This forces you to follow a standard
   (forces you to use certain methods) and in doing so gives you a channel for
   polymorphism. This isn't the only way you can do polymorphism but this is one of the
   ways.

   public class Fish implements Animal
   {
   }

40) What's the difference between the == operator and the equals() method? What test does Object.equals() use, and why?

The == operator would be used, in an object sense, to see if the two objects were
  actually the same object. This operator looks at the actually memory address to see if it
  actually the same object. The equals() method is used to compare the values of the
  object respectively. This is used in a higher level to see if the object values are equal.
  Of course the the equals() method would be overloaded in a meaningful way for
  whatever object that you were working with.

41) why do you create interfaces, and when MUST you use one.
You would create interfaces when you have two or more functionalities talking to each other. Doing it this way help you in creating a protocol between the parties involved.

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

43) How many methods do u implement if implement the Serializable Interface?

The Serializable interface is just a "marker" interface, with no methods of its own to implement.
Are there any other 'marker' interfaces?
java.rmi.Remote
java.util.EventListener

44) *What are the advantages of developing an n-tiered system?

45) *Why is it often difficult to separate the business layer from the data access layer?

46) . Diff between ArrayList and Vector

47) Variable shadowing with example

1. What is the diffrence between an Abstract class and Interface ?
2. What is user defined exception ?
3. What do you know about the garbate collector ?
4. What is the difference between C++ & Java ?
5. Explain RMI Architecture?
6. How do you communicate in between Applets & Servlets ?
7. What is the use of Servlets ?
8. What is JDBC? How do you connect to the Database ?
9. In an HTML form I have a Button which makes us to open another page in 15 seconds. How will do you that ?
10. What is the difference between Process and Threads ?
11. What is the difference between RMI & Corba ?
12. What are the services in RMI ?
13. How will you initialize an Applet ?
14. What is the order of method invocation in an Applet ?
15. When is update method called ?
16. How will you pass values from HTML page to the Servlet ?
17. Have you ever used HashTable and Dictionary ?
 18. How will you communicate between two Applets ?
19. What are statements in JAVA ?
20. What is JAR file ?
21. What is JNI ?
22. What is the base class for all swing components ?
23. What is JFC ?
24. What is Difference between AWT and Swing ?
25. Considering notepad/IE or any other thing as process, What > will happen if you start notepad or IE 3 times? Where 3 processes are started or 3 threads are started ?
26. How does thread synchronization occurs inside a monitor ?
27. How will you call an Applet using a Java Script function ?
28. Is there any tag in HTML to upload and download files ?
29. Why do you Canvas ?
30. How can you push data from an Applet to Servlet ?
31. What are 4 drivers available in JDBC ?
32. How you can know about drivers and database information ?
33. If you are truncated using JDBC, How can you know ..that how much > > > data is truncated ?
34. And What situation , each of the 4 drivers used ?
35. How will you perform transaction using JDBC ?
36. In RMI, server object first loaded into the memory and then the stub reference is sent to the client ? or whether a stub reference is directly sent to the client ?
37. Suppose server object is not loaded into the memory, and theclient request for it , what will happen?
38. What is serialization ?

39. Can you load the server object dynamically? If so, what are the major 3 steps involved in it ?
40. What is difference RMI registry and OSAgent ?
41. To a server method, the client wants to send a value 20, with this value exceeds to 20,. a message should be sent to the client ? What will you do for achieving for this ?
42. What are the benefits of Swing over AWT ?
43. Where the CardLayout is used ?
44. What is the Layout for ToolBar ?
45. What is the difference between Grid and GridbagLayout ?
46. How will you add panel to a Frame ?
47. What is the corresponding Layout for Card in Swing ?
48. What is light weight component ?
49. Can you run the product development on all operating systems ?
50. What is the webserver used for running the Servlets ?
51. What is Servlet API used for conneting database ?
52. What is bean ? Where it can be used ?
53. What is difference in between Java Class and Bean ?
54. Can we send object using Sockets ?
55. What is the RMI and Socket ?
56. How to communicate 2 threads each other ?
57. What are the files generated after using IDL to Java Compilet ?
58. What is the protocol used by server and client ?
59. Can I modify an object in CORBA ?
60. What is the functionality stubs and skeletons ?
61. What is the mapping mechanism used by Java to identify IDL language ?
62. Diff between Application and Applet ?
63. What is serializable Interface ?
64. What is the difference between CGI and Servlet ?
65. What is the use of Interface ?
66. Why Java is not fully objective oriented ?
67. Why does not support multiple Inheritance ?
68. What it the root class for all Java classes ?
69. What is polymorphism ?
70. Suppose If we have variable ' I ' in run method, If I can create one or more thread each thread will occupy a separate copy or same variable will be shared ?
71. In servlets, we are having a web page that is invoking servlets username and password ? which is cheks in the database ? Suppose the second page also If we want to verify the same information whether it will connect to the database or it will be used previous information?
72. What are virtual functions ?
73. Write down how will you create a binary Tree ?
74. What are the traverses in Binary Tree ?
75. Write a program for recursive Traverse ?
76. What are session variable in Servlets ?
77. What is client server computing ?
78. What is Constructor and Virtual function? Can we call Virtual funciton in a constructor ?
79. Why we use OOPS concepts? What is its advantage ?
80. What is the middleware ? What is the functionality of Webserver ?
81. Why Java is not 100 % pure OOPS ? (EcomServer)
82. When we will use an Interface and Abstract class ?
83. What is an RMI?
84. How will you pass parameters in RMI ? Why u serialize?
85. What is the exact difference in between Unicast and Multicast object ? Where we will use ?
86. What is the main functionality of the Remote Reference Layer ?
87. How do you download stubs from a Remote place ?
88. What is the difference in between C++ and Java ? can u explain in detail ?
89. I want to store more than 10 objects in a remote server ? Which methodology will follow ?
90. What is the main functionality of the Prepared Statement ?

91. What is meant by static query and dynamic query ?
92. What are the Normalization Rules ? Define the Normalization ?
93. What is meant by Servelet? What are the parameters of the service method ?
94. What is meant by Session ? Tell me something about HTTPSession Class ?
95. How do you invoke a Servelt? What is the difference in between doPost and doGet methods ?
96. What is the difference in between the HTTPServlet and Generic Servlet ? Expalin their methods ? Tell me their parameter names also ?
97. Have you used threads in Servelet ?
98. Write a program on RMI and JDBC using StoredProcedure ?
99. How do you sing an Applet ?
100. In a Container there are 5 components. I want to display the all the components names, how will you do that one ?
101. Why there are some null interface in java ? What does it mean ? Give me some null interfaces in JAVA ?
102. Tell me the latest versions in JAVA related areas ?
103. What is meant by class loader ? How many types are there? When will we use them ?
104. How do you load an Image in a Servlet ?
105. What is meant by flickering ?
106. What is meant by distributed Application ? Why we are using that in our applications ?
107. What is the functionality of the stub ?
108. Have you used any version control ?
109. What is the latest version of JDBC ? What are the new features are added in that ?
110. Explain 2 tier and 3 -tier Architecture ?
111. What is the role of the webserver ?
112. How have you done validation of the fileds in your project ?
113. What is the main difficulties that you are faced in your project ?
114. What is meant by cookies ? Explain ?

Make sure you have a copy of your resume in front of you. OK to have a cheat sheet or two - just don't let anyone hear papers shuffling.

Know your OOA&D definitions, such as polymorphism, inheritance, etc.
Know the difference between an interface and an abstract class.
Know that Java does not support multiple inheritance the way C++ does.
Know that you implement an interface (can implement more than one).
Know that you extend an abstract class (can only extend more than one).
Know about the access modifiers: public/friendly(default)/protected/private. Be able to explain in one or two sentences for each case. This is where you can get tangled up in a phone conversation and confuse the heck out of the interviewer and yourself.
Know AWT Event Model - tough to do over the phone - but you may get hit on a question.
Know the two ways to start a thread - "extending Thread" or "implementing Runnable".
Know that the method is "run" but to run a thread you use "start".

1. How can I improve the performance of a java application, what are the java optimization techniques.

That question is about as meaningful as "how do you code stuff?" There's no one right answer, and the interviewer probably just wanted to see if you knew *anything* about optimization. The answer is to go to the Performance thread and read the postings. If you search there for "books", you should find some good references which will get you started. You should also search for "tools".
quote:

2. What should you do to ensure that your applet works exactly the same way on both IE and netscape.

Short answers include: make sure they use the same JVM, don't use complex GUIs, and most importantly test to be certain! Personally, I'd question the benefit of the applet and ask if could be done with JSPs.