JavaEra.com®

A Perfect Place For Java Learning and Certification

CoreJava | Jsp | Servlets | JDBC | Struts | Hibernate | Spring

Projects | FAQs | Sample Resumes | eBooks | SCJP | SCWCD

Certification Stuff | Questions Banks | Tutorials | Sample Resumes

Interview Tips | Communities | Forums | Discussions

www.JavaEra.com

Q1. What is the difference between an Abstract class and Interface?

1. Abstract classes may have some executable methods and methods left unimplemented. Interfaces contain no implementation code.
2. An class can implement any number of interfaces, but subclass at most one abstract class.
3. An abstract class can have nonabstract methods. All methods of an interface are abstract.
4. An abstract class can have instance variables. An interface cannot.
5. An abstract class can define constructor. An interface cannot.
6. An abstract class can have any visibility: public, protected, private or none (package). An interface's visibility must be public or none (package).
7. An abstract class inherits from Object and includes methods such as clone() and equals().

Q2.What are checked and unchecked exceptions?

Java defines two kinds of exceptions :

• Checked exceptions : Exceptions that inherit from the Exception class are
checked exceptions. Client code has to handle the checked exceptions thrown by
the API, either in a catch clause or by forwarding it outward with the throws clause. Examples - SQLException, IOxception.

• Unchecked exceptions : RuntimeException also extends from Exception. However, all of the exceptions that inherit from RuntimeException get special treatment. There is no requirement for the client code to deal with them, and hence they are called unchecked exceptions. Example Unchecked exceptions are NullPointerException, OutOfMemoryError, DivideByZeroException typically, programming errors.

Q3.What is a user defined exception?

User-defined exceptions may be implemented by

• defining a class to respond to the exception and

• embedding a throw statement in the try block where the exception can occur or declaring that the method throws the exception (to another method where it is handled).

The developer can define a new exception by deriving it from the Exception class as follows: public class MyException extends Exception {
/* class definition of constructors (but NOT the exception handling code) goes here public MyException() {
super();
}
public MyException(String errorMessage) {
super(errorMessage);
}

}
The throw statement is used to signal the occurance of the exception within a try block. Often,
exceptions are instantiated in the same statement in which they are thrown using the syntax.
throw new MyException("I threw my own exception.")
To handle the exception within the method where it is thrown, a catch statement that handles
MyException, must follow the try block. If the developer does not want to handle the exception in the method itself, the method must pass the exception using the syntax: public myMethodName() throws MyException

Q4.What is the difference between C++ & Java?
Well as Bjarne Stroustrup says "..despite the syntactic similarities, C++ and Java are very
different languages. In many ways, Java seems closer to Smalltalk than to C++..". Here are few I discovered:
• Java is multithreaded
• Java has no pointers
• Java has automatic memory management (garbage collection)
• Java is platform independent (Stroustrup may differ by saying "Java is a platform"
• Java has built-in support for comment documentation
• Java has no operator overloading
• Java doesn't provide multiple inheritance
• There are no destructors in Java

Q5.What are statements in JAVA ?
Statements are equivalent to sentences in natural languages. A statement forms a complete unit of execution. The following types of expressions can be made into a statement by terminating the expression with a semicolon
• Assignment expressions
• Any use of ++ or --
• Method calls
• Object creation expressions
These kinds of statements are called expression statements. In addition to these kinds of
expression statements, there are two other kinds of statements. A declaration statement
declares a variable. A control flow statement regulates the order in which statements get
executed. The for loop and the if statement are both examples of control flow statements.

Q6.What is JAR file?
JavaARchive files are a big glob of Java classes, images, audio, etc.,

compressed to make one simple, smaller file to ease Applet downloading. Normally when a browser encounters an applet, it goes and downloads all the files, images, audio, used by the Applet separately. This can lead to slower downloads.

Q7.What is JNI?

JNI is an acronym of Java Native Interface. Using JNI we can call functions which are written in other languages from Java. Following are its advantages and disadvantages.

Advantages:

• You want to use your existing library which was previously written in other language.
• You want to call Windows API function.
• For the sake of execution speed.
• You want to call API function of some server product which is in c or c++ from java client.

Disadvantages:

• You can't say write once run anywhere.
• Difficult to debug runtime error in native code.
• Potential security risk.
• You can't call it from Applet.

Q8.What is serialization?

Quite simply, object serialization provides a program the ability to read or write a whole object
to and from a raw byte stream. It allows Java objects and primitives to be encoded into a byte stream suitable for streaming to some type of network or to a file-system, or more generally, to a transmission medium or storage facility. A seralizable object must implement the Serilizable
interface. We use ObjectOutputStream to write this object to a stream and ObjectInputStream to read it from the stream.

Q9.Why there are some null interface in java ? What does it mean? Give me some null interfaces in JAVA?

Null interfaces act as markers..they just tell the compiler that the objects of this class need to be treated differently..some marker interfaces are : Serializable, Remote, Cloneable.

Q10. Is synchronized a modifier? Identifier? What is it??

It's a modifier. 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.

Q11.What is singleton class? where is it used?
Singleton is a design pattern meant to provide one and only one instance of an object. Other
objects can get a reference to this instance through a static method (class constructor is kept
private). Why do we need one? Sometimes it is necessary, and often sufficient, to create a single instance of a given class. This has advantages in memory management, and for Java, in garbage collection. Moreover, restricting the number of instances may be necessary or desirable for
technological or business reasons--for example, we may only want a single instance of a pool of database connections.

Q12.What is a compilation unit?
The smallest unit of source code that can be compiled, i.e. a .java file.

Q13.Is strings a wrapper class?
String is a class, but not a wrapper class. Wrapper classes like (Integer) exist for each primitive type. They can be used to convert a primitive data value into an object, and vice versa.

Q14.Why java does not have multiple inheritances?
The Java design team strove to make Java:
• Simple, object oriented, and familiar
• Robust and secure
• Architecture neutral and portable
• High performance
• Interpreted, threaded, and dynamic
The reasons for omitting multiple inheritance from the Java language mostly stem from the
"simple, object oriented, and familiar" goal. As a simple language, Java's creators wanted a
language that most developers could grasp without extensive training. To that end, they worked to make the language as similar to C++ as possible (familiar) without carrying over C++'s unnecessary complexity (simple).
In the designers' opinion, multiple inheritance causes more problems and confusion than it
solves. So they cut multiple inheritance from the language (just as they cut operator
overloading). The designers' extensive C++ experience taught them that multiple inheritance just wasn't worth the headache.

Q15.Why java is not a 100% oops?
Many people say this because Java uses primitive types such as int, char, double. But then all the rest are objects. Confusing question.

Q16.What is a resource bundle?

In its simplest form, a resource bundle is represented by a text file containing keys and a text value for each key.
* Core Java
* Servlets & JSP
* Struts
* EJB
* J2ME

Q17.What is transient variable?
Transient variable can't be serialize. For example if a variable is declared as transient in a
Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStream the value of the variable becomes null.

Q18.What is Collection API?
The Collection API is a set of classes and interfaces that support operation on collections of
objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.
Example of classes: HashSet, HashMap, ArrayList, LinkedList, TreeSet and TreeMap. Example of interfaces: Collection, Set, List and Map.

Q19.Is Iterator a Class or Interface? What is its use?
Iterator is an interface which is used to step through the elements of a Collection.

Q20.What is similarities/difference between an Abstract class and Interface?
Differences are as follows:
• Interfaces provide a form of multiple inheritance. A class can extend only one other class.
• Interfaces are limited to public methods and constants with no implementation.
Abstract classes can have a partial implementation, protected parts, static methods,etc.
• A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.
• Interfaces are slow as it requires extra indirection to to find corresponding method in in the actual class. Abstract classes are fast.
Similarities:
• Neither Abstract classes or Interface can be instantiated.

Q21.What is a transient variable?
A transient variable is a variable that may not be serialized.

Q22.Which containers use a border Layout as their defaultlayout?

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

Q23.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.

Q24.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.

Q25.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.

Q26. 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.

Q27. 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.

Q28. Is null a keyword?
The null value is not a keyword.

Q29. 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.

Q30. What method is used to specify a container's layout?
The setLayout() method is used to specify a container's layout.

Q31. Which containers use a FlowLayout as their defaultlayout?
The Panel and Applet classes use the FlowLayout as their default layout.

Q32. What state does a thread enter when it terminates itsprocessing?
When a thread terminates its processing, it enters the dead state.

Q33. What is the Collections API?
The Collections API is a set of classes and interfaces that support operations on collections of objects.

Q34. Which characters may be used as the second character of an identifier, but not as the first character of anidentifier?
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.

Q35. What is the List interface?
The List interface provides support for ordered collections of objects.

Q36. How does Java handle integer overflows andunderflows?
It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

Q37. What is the Vector class?
The Vector class provides the capability to implement a growable array of objects

Q38. What modifiers may be used with an inner class that isa member of an outer class?
A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

Q39. What is an Iterator interface?
The Iterator interface is used to step through the elements of a Collection.

Q40. 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.

Q41. Which method of the Component class is used to setthe position and size of a component?
setBounds()

Q42. 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.

Q43. 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.

Q44. Which java.util classes and interfaces support eventhandling?
The EventObject class and the EventListener interface support event processing.

Q45. Is sizeof a keyword?
The sizeof operator is not a keyword.

Q46. What are wrapped classes?
Wrapped classes are classes that allow primitive types to be accessed as objects.

Q47. Does garbage collection guarantee that a program willnot 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

Q48. What restrictions are placed on the location of apackage 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).

Q49. Can an object's finalize() method be invoked while it isreachable?
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.

Q50. What is the immediate superclass of the Applet class?
Panel

Q51. What is the difference between preemptive schedulingand 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.

Q52 Name three Component subclasses that supportpainting.
The Canvas, Frame, Panel, and Applet classes support painting.

Q53. What value does readLine() return when it hasreached the end of a file?
The readLine() method returns null when it has reached the end of a file.

Q54. What is the immediate superclass of the Dialog class?
Window

Q55. What is clipping?
Clipping is the process of confining paint operations to a limited area or shape.

Q56. What is a native method?
A native method is a method that is implemented in a language other than Java.

Q57. Can a for statement loop indefinitely?
Yes, for statement can loop indefinitely. For example, consider the following: for(; ;

Q58. What are order of precedence and associativity, andhow 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

Q59. When a thread blocks on I/O, what state does itenter?
A thread enters the waiting state when it blocks on I/O.

Q60. To what value is a variable of the String typeautomatically initialized?
The default value of a String type is null.

Q61. What is the catch or declare rule for methoddeclarations?
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.

Q62. What is the difference between a MenuItem and aCheckboxMenuItem?
The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.

Q63. What is a task's priority and how is it used inscheduling?
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.

Q64. 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.

Q65. When a thread is created and started, what is its initialstate?
A thread is in the ready state after it has been created and started.

Q66. Can an anonymous class be declared as implementingan interface and extending a class?
An anonymous class may implement an interface or extend a superclass, but may not be declared to do both.

Q67. What is the range of the short type?
The range of the short type is -(2^15) to 2^15 - 1.

Q68. What is the range of the char type?
The range of the char type is 0 to 2^16 - 1.

Q69. In which package are most of the AWT events thatsupport 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.

Q70. What is the immediate superclass of Menu?
MenuItem

Q71. 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.

Q72. Which class is the immediate superclass of theMenuComponent class.
Object

Q73. 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.

Q74. What is the difference between the Boolean & operatorand 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.

Q75. Name three subclasses of the Component class.
Box.Filler, Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, or TextComponent.

Q76. What is the GregorianCalendar class?
The GregorianCalendar provides support for traditional Western calendars.

Q77. Which Container method is used to cause a containerto be laid out and redisplayed?
validate()

Q78. What is the purpose of the Runtime class?
The purpose of the Runtime class is to provide access to the Java runtime system.

Q79. How many times may an object's finalize() method beinvoked by the garbage collector?
An object's finalize () method may only be invoked once by the garbage collector.

Q80. What is the purpose of the finally clause of a try-catchfinally statement?
The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

Q81. What is the argument type of a program's main()method?
A program's main () method takes an argument of the String[] type.

Q82. Which Java operator is right associative?

The = operator is right associative.

Q83. 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.

Q84. Can a double value be cast to a byte?
Yes, a double value can be cast to a byte.

Q85. What is the difference between a break statement anda 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.

Q86. 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.

Q87. What method is invoked to cause an object to beginexecuting as a separate thread?
The start() method of the Thread class is invoked to cause an object to begin executing as a separate thread.

Q88. Name two subclasses of the TextComponent class.
TextField and TextArea

Q89. What is the advantage of the event-delegation modelover the earlier eventinheritance 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.

Q90. Which containers may have a MenuBar?
Frame

Q91. How are commas used in the intialization and iterationparts of a for statement?
Commas are used to separate multiple statements within the initialization

and iteration parts of a for statement.

Q92. What is the purpose of the wait (), notify (), andnotifyAll() 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..

Q93. What is an abstract method?
An abstract method is a method whose implementation is deferred to a subclass.

Q94. 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.

Q95. What is the relationship between the Canvas class andthe Graphics class?
A Canvas object provides access to a Graphics object via its paint() method.

Q96. What are the high-level thread states?
The high-level thread states are ready, running, waiting, and dead.

Q97. What value does read() return when it has reached theend of a file?
The read() method returns -1 when it has reached the end of a file.

Q98. Can a Byte object be cast to a double value?
No, an object cannot be cast to a primitive value.

Q99. What is the difference between a static and a nonstaticinner 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.

Q100. What is the difference between the String andStringBuffer classes?
String objects are constants. StringBuffer objects are not.

Q101. If a variable is declared as private, where may thevariable be accessed?
A private variable may only be accessed within the class in which it is declared.

Q102. What is an object's lock and which object's havelocks?
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.

Q103. What is the Dictionary class?
The Dictionary class provides the capability to store key-value pairs.

Q104. 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.

Q105. 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.

Q106. When can an object reference be cast to an interfacereference?
An object reference be cast to an interface reference when the object implements the referenced interface.
Q107. What is the difference between a Window and aFrame?
The Frame class extends Window to define a main application window that can have a menu bar.

Q108. Which class is extended by all other classes?
The Object class is extended by all other classes.

Q109. Can an object be garbage collected while it is stillreachable?
A reachable object cannot be garbage collected. Only unreachable objects may be garbage collected..

Q110. Is the ternary operator written x : y ? z or x ? y : z ?
It is written x ? y : z.

Q111. What is the difference between the Font andFontMetrics classes?
The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object.

Q112. How is rounding performed under integer division?
The fractional part of the result is truncated. This is known as rounding toward zero.

Q113. 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.

Q114. 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.

Q115. 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.

Q116. 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.

Q117. What is the SimpleTimeZone class?
The SimpleTimeZone class provides support for a Gregorian calendar.

Q118. What is the Map interface?
The Map interface replaces the JDK 1.1 Dictionary class and is used associate keys with values.

Q119. Does a class inherit the constructors of itssuperclass?
A class does not inherit constructors from any of its superclasses.

Q120. For which statements does it make sense to use alabel?
The only statements for which it makes sense to use a label are those statements that can enclose a break or continue statement.

Q121. What is the purpose of the System class?

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

Q122. Which TextComponent method is used to set aTextComponent to the read-only state?
setEditable()

Q123. 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.

Q124. Is &&= a valid Java operator?
No, it is not.

Q125. Name the eight primitive Java types?
The eight primitive types are byte, char, short, int, long, float, double, and boolean.

Q126. Which class should you use to obtain designinformation about an object?
The Class class is used to obtain information about an object's design.

Q127. What is the relationship between clipping andrepainting?
When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.

Q128. Is "abc" a primitive value?
The String literal "abc" is not a primitive value. It is a String object.

Q129. What is the relationship between an event-listenerinterface 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 eventlistener interface.

Q130. What restrictions are placed on the values of eachcase 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.

Q131. What modifiers may be used with an interfacedeclaration?
An interface may be declared as public or abstract.

Q132. Is a class a subclass of itself?
A class is a subclass of itself.

Q133. What is the highest-level event class of the event delegationmodel?

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

Q134. What event results from the clicking of a button?
The ActionEvent event is generated as the result of the clicking of a button.

Q135. 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.

Q136. What is the difference between a while statementand 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.

Q137. 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.

Q138. What advantage do Java's layout managers provideover 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.

Q139. 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.

Q140. What modifiers can be used with a local inner class?
A local inner class may be final or abstract.

Q141. What is the difference between static and non-staticvariables?
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.

Q142. 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.

Q143. 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.

Q144. Can an exception be rethrown?
Yes, an exception can be rethrown.

Q145. Which Math method is used to calculate the absolutevalue of a number?
The abs() method is used to calculate absolute values.

Q146. 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.

Q147. When does the compiler supply a default constructorfor a class?
The compiler supplies a default constructor for a class if no other constructors are provided.

Q148. When is the finally clause of a try-catch-finallystatement 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.

Q149. Which class is the immediate superclass of theContainer class?
Component

Q150. If a method is declared as protected, where may themethod 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.

Q151. How can the Checkbox class be used to create a radiobutton?
By associating Checkbox objects with a CheckboxGroup.

Q152. Which non-Unicode letter characters may be used asthe first character of an identifier?
The non-Unicode letter characters $ and _ may appear as the first character of an identifier

Q153. What restrictions are placed on method overloading?
Two methods may not have the same name and argument list but different return types.

Q154. What happens when you invoke a thread's interruptmethod 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.

Q155. 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.

Q156. What is the return type of a program's main()method?
A program's main() method has a void return type.

Q157. Name four Container classes.
Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

Q158. 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.

Q159. What class of exceptions are generated by the Javarun-time system?

The Java runtime system generates RuntimeException and Error exceptions.

Q160. What class allows you to read objects directly from astream?
The ObjectInputStream class supports the reading of objects from input streams.

Q161. What is the difference between a field variable and alocal 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.

Q162. 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.

Q163. 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.

Q164. What is the relationship between a method's throwsclause and the exceptions that can be thrown during themethod's execution?
A method's throws clause must declare any checked exceptions that are not caught within the body of the method.

Q165. What is the difference between the JDK 1.02 eventmodel and the event-delegation model introduced withJDK 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.

Q166. How is it possible for two String objects withidentical

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.

Q167. Why are the methods of the Math class static?
So they can be invoked as if they are a mathematical code library.

Q168. What Checkbox method allows you to tell if aCheckbox is checked?
getState()
Q169. What state is a thread in when it is executing?
An executing thread is in the running state.

Q170. What are the legal operands of the instanceofoperator?
The left operand is an object reference or null value and the right operand is a class, interface, or array type.

Q171. 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.

Q172. 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.

Q173. If an object is garbage collected, can it becomereachable again?
Once an object is garbage collected, it ceases to exist. It can no longer become reachable again.

Q174. 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.

Q175. What classes of exceptions may be thrown by a throwstatement?
A throw statement may throw any expression that may be assigned to the Throwable type.

Q176. What are E and PI?
E is the base of the natural logarithm and PI is mathematical value pi.

Q177. Are true and false keywords?
The values true and false are not keywords.

Q178. What is a void return type?
A void return type indicates that a method does not return a value.

Q179. 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 eventdispatch methods.

Q180. What is the difference between the File andRandomAccessFile 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.

Q181. What happens when you add a double value to aString?
The result is a String object.

Q182. 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..

Q183. Which package is always imported by default?
The java.lang package is always imported by default.

Q184. What interface must an object implement before itcan 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.

Q185. 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.

Q186. 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.

Q187. What is a compilation unit?
A compilation unit is a Java source code file.

Q188. What interface is extended by AWT event listeners?
All AWT event listeners extend the java.util.EventListener interface.

Q189. 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 thrownby the overridden method.

Q190. How can a dead thread be restarted?
A dead thread cannot be restarted.

Q191. 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.

Q192. What is a layout manager?
A layout manager is an object that is used to organize components in a container.

Q193. Which arithmetic operations can result in thethrowing of an ArithmeticException?
Integer / and % can result in the throwing of an ArithmeticException.

Q194. What are three ways in which a thread can enter thewaiting 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.

Q195. Can an abstract class be final?
An abstract class may not be declared as final.

Q196. 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.

Q197. What happens if a try-catch-finally statement doesnot 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.

Q198. 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.

Q199. What is the difference between a Scrollbar and aScrollPane?
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.

Q200. What is the difference between a public and a nonpublicclass?
A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

Q201. To what value is a variable of the boolean typeautomatically initialized?
The default value of the boolean type is false.

Q202. Can try statements be nested?
Try statements may be tested.

Q203. What is the difference between the prefix and postfixforms 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.

Q204. What is the purpose of a statement block?
A statement block is used to organize a sequence of statements as a single statement group.

Q205. 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.

Q206. What modifiers may be used with a top-level class?
A top-level class may be public, abstract, or final.

Q207. 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..

Q208. 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.

Q209. 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.

Q210. 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.

Q211. 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.

Q212. What methods are used to get and set the text labeldisplayed by a Button object?
getLabel() and setLabel()

Q213. Which Component subclass is used for drawing andpainting?
Canvas

Q214. What are synchronized methods and synchronizedstatements?
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.

Q215. What are the two basic ways in which classes thatcan 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.

Q216. What are the problems faced by Java programmerswho 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.

Q217. What is the difference between an if statement and aswitch 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.

Q218. What happens when you add a double value to aString?
The result is a String object.

Q219. What is the List interface?
The List interface provides support for ordered collections of objects

Q220.What is an Exception?
Ans.An unwanted, unexpected event that disturbs normal flow of the program is called Exception.Example: FileNotFondException.

Q221.What is the purpose of Exception Handling?
Ans.The main purpose of Exception Handling is for graceful termination of the program.

Q222.What is the meaning of Exception Handling?
Ans. Exception Handling doesn’t mean repairing an Exception, we have to define alternative way to continue rest of the code normally.
Example: If our programming requirement is to read the data from the file locating at London but at Runtime if London file is not available then we have to use local file alternatively to continue rest of program normally. This is nothing but Exception Handling.

Q223.Explain Default Exception Handling Mechanism in java?

Ans.If an exception raised, the method in which it's raised is responsible for the creation of Exceptions object by including the following information:
Name of the Exception
Description of the Exception
Stack Trace
After creating Exception object the method handover it to the JVM. JVM checks for Exception Handling code in that method.
If the method doesn't contain any Exception handling code then JVM terminates the
method abnormally and removes the corresponding entry from the stack.
JVM identify the caller method and checks for Exception Handling code in that method. If the caller doesn't contain any exception handling code then JVM terminates that method abnormally and removes the corresponding entry from the stack.
This process will be continue until main() method.
If the main() method also doesn't contain exception handling code the JVM terminates that main() method and removes the corresponding entry from the stack. Just before terminating the program abnormally JVM handovers the responsibility of
exception handling to the Default Exception Handler which is the component of JVM.
Default Exception Handler just print exception information to the consol in the following format

Name of Exception: Description
Stack Trace (Location of the Exception)

Q224.What is the purpose of try?
Ans We should maintain all risky code inside the try block.

Q225. What is the purpose of catch block?
Ans.We have to maintain all Exception Handling code inside the catch block.

Q226. Is try with multiple catch block is possible?
Ans. The way of handling an exception is varied from exception to exception compulsory we have to write a separate catch block for every exception. Hence try will multiple catch block is possible and it is recommended to use.
Example:

```
try{
//Risky code
}
catch(IOException e)
{
//Hndling code for IOException
```

```
}
catch(ArithmeticException e) {
//handling code for AE
}
catch(NullPointerExcetpion e)
{
// handling code for NPE
}
catch(Exception e)
{
//default exception handling code
}
```

Q227. If try with multiple catch block present is order of catch blocks important in which order we have to take?

Ans. If try with multiple catch block present then the order of catch block is very important it should be from child to parent but not from parent to child.

Q228. What are various methods to print Exception information? and differentiate them.

Ans.

Throwable class defines the following method to print exception or error information .

1. printStackTrace() :- This method print exception information in the following format.

Name of the Exception: Description

StackTrace

2.toString():- This method print exception information in the following format.

Name of the Exception: Description

3.getMessage():- This method prints only description of the exception.

Description

Q229.If an exception rised inside catch block then what will happen?

Ans. If an exception raised inside catch block and it is not part of any try block then it is always abnormal termination.

Q230. Is it possible to take try, catch inside try block?

Ans. Yes, It is possible to take try, catch inside try block. That is nesting of try catch is possible.

Q231.Is it possible to take try, catch inside catch block?

Ans. Yes, It is possible to take try, catch inside catch block.

Q232. Is it possible to take try without catch?
Ans. Yes, it is possible to take try without catch but compulsory finally block should be available.

Q233. What is the purpose of finally block?
Ans. The main purpose of finally block is, to maintain the cleanup code. This block will execute always.

Q234. Is finally block will be execute always?
Ans. Yes finally block will be executed always irrespective of whether exception raised or not raised whether exceptions are handled or not handle. There is one situation where the finally block won't be executed if the JVM is going to be shutdown.

Q235. In which situation finally block will not executed?
Ans. There is one situation where the finally block won't be executed if we are using
system.exit(0) explicitly then JVM itself will be shutdown and there is no chance of executing finally block.

Q236. If return statement present inside try is finally block will be executed?
Ans. Yes, if return statement present inside try, then also finally block will be executed. finally block will dominate return statement also.

Q237. What is the difference between final, finally and finalize()?
Ans. final:- final is a modifier applicable for variables, methods and classes. final variable
means constant and reassignment is not possible. final method means implementation is final in the child classes we can't override. final class means it won't participate in inheritance and child class creation is not possible.
finally:- It is a block associated with try catch to maintain cleanup code. Finally block will be executed always irrespective of whether exception is raised or not raised or whether the exception is handle or not handle.
finalize():- It is a method, Garbage collector always calls this method just before destroying any object to perform cleanup activities.

Q238. Is it possible to write any statement between try-catch and finally?
Ans. No, it is not possible to write any statement between try catch and finally. If we will try to write any statement between them then we will get compile time error.

Q239. Is it possible to take two finally blocks for the same try?
Ans. No, it is not possible to take two finally blocks for the same try. If we try to take then we will get compile time error.

Q240. Is syntax try-finally-catch is valid ?
Ans. No, this syntax is not valid. It should be like try-catch-finally then only code will compile.

Q241. What is the purpose of throw?
Ans. Sometimes we can create Exception object explicitly and we can handover that exception object to the JVM explicitly by throw keyword. The purpose of throw keyword is to handover our created exception object explicitly to the
JVM.
Example1:

```
class Test{
public static void main(String[] args){
System.out.println(10/0);
}
}
```

In this case ArithmeticException object created implicitly and handover to the JVM automatically by the main method.
Example2:

```
class Test{
Public static void main(String[] args){
Throw new ArithmeticException("/by Zero"); }
}
```

In this case creation of an exception object and handover to the JVM explicitly by the programmer.

Q242. Is it possible to throw an Error?
Ans. Yes, It is possible to throw any Throwable type including Error.

Q243. Is it possible to throw any java object?
Ans. No, we can use throw keyword only for throwable objects otherwise we will get compile time error saying incompatible type.
* Core Java
* Servlets & JSP
* Struts
* EJB
* J2ME

Q244. After throw is it allow to take any statement directly?

Ans. After throw statement we are not allow to place any statement directly violation leads to compile time error saying Unreachable Statement.

Q245. What is the purpose of throws?
Ans. The main purpose of throws keyword is to delegate the responsibilities of exception handling to the caller. It requires in the case of checked exception.

Q246. What is the difference between throw and throws?
Ans. Sometimes we can create Exception object explicitly and we can handover that exception object to the JVM explicitly by throw keyword.The main purpose of throw keyword is to
handover our created exception object explicitly to the JVM. The main purpose of throws
keyword is to delegate the responsibilities of exception handling to the caller. It requires in the case of checked exception.

Q47. What is the difference between throw and thrown?
Ans. There is no terminology of thrown in java.

Q248. Is it possible to use throws keyword for any java class?
Ans. No, we can use throws keyword only for Throwable classes. Otherwise we will get compile time error saying Incompatible types.

Q249. If we are taking catch block for an exception but there is no chance of rising that exception in try then what will happen?
Ans. If there is no chance of raising an exception in try then we are not allow to write catch
block for that exception violation leads to compile time error. But this rule is applicable only for fully checked exception.

Q250. Explain Exception Handling keyword?
Ans. Exception Handling keyword: Try :- To maintain Risky code.
Catch:- To maintain Exception Handling code. Finally:- To maintain the clean up code.
Throw:- To handover our created exception object to the JVM explicitly.
Throws:- To delegate the responsibilities of Exception Handling to the caller.

Q251. Which class act as root for entire java Exception hierarchy?
Ans. Throwable class act as root for entire java Exception hierarchy.

Q252. What is the difference between Error and Exception?
Ans. Throwable class contain two child classes.
Exception:- These are mostly caused by our program and are recoverable.

Error:- These are not caused by our program, mostly caused by lake of system resources. These are non recoverable.

Q253. What is difference between checked exception and unchecked exception?

Ans. The exceptions which are checked by the compiler for smooth execution of the program at Runtime is called checked exception. Example: IOException, InterruptedException.The
exceptions which are not checked by the compiler are called unchecked exception. Example: ArithmeticException,RuntimeException.

Q254.What is difference between partially checked and fully checked Exception?

Ans. A checked exception is said to be fully checked if and only if all the child classes also checked otherwise it is called partially checked exception.
Example:
IOException:- fully checked exception
Exception:- partially checked exception
Throwable:- partially checked exception
RuntimeException:- unchecked exception

Q255. What is a customized Exception?

Ans. Sometimes based on our programming requirement we have to create our own exception such type of exception are called customize Exception.
Example:
TooYoungException
TooOldException
InsufficientFundException

Q256. Explain the process of creating the customized Exception.

Ans. Creating customized Exception:

```
Class TooYoungException extends RuntimeException{
TooYoungExcetpion(String desc){
Super(desc);
}
}
Class TooOldException extends RuntimeException {
TooOldException(String desc){
super(desc);
}
}
Class custExcepiton{
Public static void main(String[] args){ int age=Integer.parseInt(args[0]);
if(age>60)
```

```
{
Throw new TooYoungException("Please wait some more time, definitely
you will get best match");
}
else if(age<18) {
Throw new TooOldException("Your age is already crossed of marriage, no
chance to getting marriage");
}
else {
System.out.println("Congratulation! You will get match details soon by
your
email");
}
}
```

Q257. Explain control flow in try, catch, finally.

Ans.
```
try{
Statement1;
Statement2;
Statement3;
}
catch(X e){
Statement4; }
Finally{
Statement5; }
Statement6;
```

Case1:
If there is no Exception then output is
Statement1
Statement2
Statement3
Statement5
Statement6
Normal termination

Case2:
If an exception raised at statement2 and corresponding catch block has matched then output is Statement1
Statement4
Statement5
Statement5
Normal termination

Case3:
An exception raised at statement2 and corresponding catch has not matched then output is Statement1
Statement5
Abnormal termination

Case4:

An exception occurs at statement4 it always Abnormal termination but before that finally block will be executed and output is
Statement1
Statement2
Statement5
Abnormal termination
Case5:
If an exception raised at statement5 or statement6, it is always abnormal termination.

Q258. Can you give the most common occurred exception in your previous project.
Ans. NullPointerException, ArrayIndexOutofBoundException, StackOverFlowError,
ClassCastException, NoClassDefFoundError, ExceptionInitilizerError, IllegalArgumentException, NumberFormatException, IllegalStateException, AssertionError.

Q259. What is Multitasking?
Ans. Executing several task simultaneously is called multitasking.

Q260. What is the difference between process-based and Thread-based Multitasking?
Ans.Process-based multitasking:- Executing several task simultaneously where each task is a
separate independent process such type of multitasking is called process based Multitasking.
Example:-While typing a program in the editor we can listen MP3 audio songs. At the same time we download a file from the net. all these task are executing simultaneously and each task is
a separate independent program. hence it is process based multitasking. It is best suitable at operating system level. Thread-based multitasking:- Executing several task simultaneously where each task is a separate independent part of the same program is called Thread-based multitasking. and every independent part is called a thread. This type of multitasking is best suitable at programmatic level.

Q261. What is Multithreading and explain its application areas?
Ans. Executing several thread simultaneously where each thread is a separate independent part of the same program is called multithreading. Java language provides inbuilt support for
multithreading by defining a reach library, classes and interfaces like Thread, ThreadGroup,
Runnable etc. The main important application area of multithreading are video games implementation, animation development, multimedia graphics etc.

Q262.What is advantage of Multithreading?

Ans. The main advantage of multithreading is reduces response time and improves performance of the system.

Q263. When compared with C++ what is the advantage in java with respect to Multithreading?

Ans.Java language provides inbuilt support for multithreading by defining a reach library, classes and interfaces like Thread, ThreadGroup, Runnable etc. But in c++ there is no inbuilt support for multithreading.

Q264. In how many ways we can define a Thread? Among extending Thread and implementingRunnable which is recommended?

Ans. We can define a Thread in the following two ways:
1. by extending Thread class or
2. by implementing Runnable interface.

Among the two ways of defining a thread implementing Runnable mechanism is always
recommended. In the first approach as our Thread class already extending Thread there is no chance of extending any other. Hence, we missing the key benefit of oops(inheritance properties).

Q265. What is the difference between t.start() and t.run() method?

Ans. In the case of t.start() method, a new thread will be created which is responsible for the
execution of run() method.
But in the case of t.run() method no new thread will be created main thread executes
run() method just like a normal method call.

Q266. Explain about Thread Scheduler?

Ans. If multiple threads are waiting for getting the chance for executing then which thread will get chance first decided by Thread Scheduler. It is the part of JVM and its behavior is vendor dependent and we can't expect exact output.Whenever the situation comes to multithreading the guarantee behavior is very- very low.

Q267. If we are not overriding run() method what will happened?

Ans.If we are not overriding run() method then Thread class run() method will executed which has empty implementation and hence we will not get any output.

Q268.Is overloading of run() method is possible?

Ans.Yes, we can overload run() method but Thread class start() method

always invokes no-
argument run() method only. The other run() method we have to call explicitly then only will be executed.

Q269.Is it possible to override start() method?
Ans. Yes it is possible. But not recommended.

Q270.If we are overriding start() method then what will happen?
Ans. It we are overriding start() method then our own start() method will be executed just like a normal method call. In this case no new Thread will be created.

Q271. Explain life cycle of a Thread?
Ans. Once we create a Thread object then the Thread is said to be in New/Born state once we call t.start() method now the Thread will be entered into ready/Runnable state that is Thread is ready to execute. If Thread Scheduler allocates CPU now the Thread will entered into the Running state and start execution of run() method. After completing run() method the Thread entered into Dead State.

Q272. What is the importance of Thread class start() method?
Ans. Start() method present in Thread class performing all low level joining formalities for the newly created thread like registering thread with Thread Scheduler etc and then start() method
invoking run() method.As the start() method is doing all low level mandatory activities,
Programmer has to concentrate only on run() method to define the job. Hence, start() method is a big assistant to the programmer.Without executing Thread class start() method there is no chance of starting a new Thread.

Q273. After starting a Thread if we trying to restart the same thread once again what will happen?
Ans. After starting a Thread restarting of the same Thread once again is not allowed violation leads to Runtime Exception saying IllegalThreadStateException.

Q274. Explain Thread class constructors?
Ans. There are eight constructors are available in Thread class:
1. Thread t=new Thread();
2. Thread t=new Thread(Runnable r);
3. Thread t=new Thread(String name);
4.Thread t=new Thread(Runnable r, String name);
5.Thread t=new Thread(ThreadGroup g, String name);
6.Thread t=new Thread(ThreadGroup g, Runnable r);

7.Thread t=new Thread(ThreadGroup g, Runnable r, String name);
8.Thread t=new Thread(ThreadGroup g, Runnable r, String name, long stacksize);

Q275. How to get and set name of a Thread?
Ans. For every Thread in java there is a name. To set and get the name of a Thread we can use the following methods. All methods are final.
1.Public final void setName(String name); - To set the name of a Thread
2.Public final String getName(); - To get the name of a Thread.

Q276. What is the range of Thread priority in java?
Ans. The valid range of a Thread priority is 1-10. (1 is least priority and 10 is highest priority)

Q277. Who uses Thread priority?
Ans. Thread Scheduler uses priorities while allocating CPU. The Thread which is having highest priority will get chance first for execution.

Q278. What is the default priority of the Thread?
Ans. The default priority only for the main thread is 5 but for all remaining threads default
priority will be inheriting from parent to child. Whatever priority parent thread has the same will be inherited to the child thread.

Q279. Once we created a new Thread what about its priority?
Ans. Whatever priority parent thread has the same will be inherited to the new child thread.

Q280. How to get and set priority of a Thread?
Ans. To get and set priority of a Thread, Thread class defines the following two methods:;
1. Public final int
getPriority();
2. Public final void setPriority(int priority);

Q281. If we are trying to set priority of a Thread as 100 what will happen?
Ans. If we are trying to set priority of a Thread as 100 then we will not get any compile time error but at the runtime we will get Runtime exception IllegalArgumentException. Because the valid range of the Thread priority is (1-10) only.

Q282. If two threads having same priority then which thread will get chance first for execution?
Ans. If two threads having same priority then which thread will get the chance first for execution decided by Thread Scheduler. It is the part of

JVM and its behavior is vendor dependent and we can't expect exact output.

Q283. If two threads having different priority then which thread will get chance first for execution?
Ans. If two threads having different priority then the Thread which is having highest priority will get chance first for execution.

Q284 .How we can prevent a thread from execution?
Ans. We can prevent a Thread from executin by using the following methods:
1. Yield()
2. Join()
3. Sleep()

Q285. What is yield() method? Explain its purpose?
Ans. yield() method causes the current executing thread to pause execution and give the chance for waiting thread are same priority. If there is no waiting thread or all the remaining waiting thread have low priority then the same thread will get chance once again for execution. The
Thread which is yielded when it will get chance once again for execution depends upon mercy of Thread scheduler.Public static native void yield();

Q286.What is purpose of join() method?
Ans. If a Thread wants to wait until some other Thread completion then we should go for join() method.
Example: if a Thread t1 execute t2.join() ; then t1 will entered into waiting state until t2 Thread completion.

Q287. Is join() method is overloaded?
Ans. Yes join() method is overloaded method.
Public final void join() throws InterruptedException
By using this method thread will wait up to another thread completion .
Public final void join(long ms) throws InterruptedException
By using this method thread will wail upto sometime what we are passing as a argument in millisecond
Public final void join(long ms, int ns)throws InterruptedException
By using this method thread will wait up to sometime what we are passing as a argument in millisecond and nanosecond.

Q288. What is the purpose of sleep() method?
Ans. If a Thread don't want to perform any operation for a particular amount of time then we should go for sleep() method.Whenever we are

using sleep() method compulsory we should
handle InterruptedException either by using try-catch or by using throws keyword otherwise we will get compile time error.

Q289. What is synchronized keyword? Explain its advantages and disadvantages.

Ans. Synchronized keyword is applicable for method and blocks only. We can't use for
variables and classes.
If a method declared as a synchronized then at a time only one Thread is allow to
execute that method on the given object.
The main advantages of synchronized keyword are, we can prevent data inconsistency problems and we can provide Threadsafty.
But the main limitation of synchronized keyword is it increases waiting time of
Threads and effect performance of the system. Hence if there is no specific requirement it is not recommended to use synchronized keyword.

Q290.Where we can use synchronized keyword?

Ans. Synchronization concept is applicable whenever multiple Threads are operating on the same object simultaneously. But whenever multiple Threads are operating on different objects then there is no impact of synchronization.

Q291. What is object lock? Explain when it is required?

Ans. Every object in java has a unique lock whenever we are using synchronization concept then only lock concept will coming to the picture. If a Thread wants to execute a synchronized method first it has to get the lock of the object. Once a Thread got the lock then it is allow to execute any synchronized method on that object. After completing synchronized method execution Thread releases the lock automatically.While a Thread executing synchronized method on the given object the remaining Threads are not allow to execute any synchronized method on that object simultaneously. But remaining Threads are allow to execute any non-synchronized method simultaneously. (Lock concept is implemented based on object but not based on method.)

Q292.What is the class level lock? Explain its purpose.

Ans. Every class in java has a unique lock if a Thread wants to execute static synchronized
method that Thread has to get class level lock once a Thread got class level lock then only it is allow to execute static synchronized method. While a Thread executing any static synchronized method then remaining Threads are not allow to execute any static synchronized method of the same class simultaneously. But the remaining Threads are allow to execute the following method simultaneously:

1. Any static non-synchronized method.
2. Synchronized instance methods
3. Non-synchronized instance method.

There is no relationship between object lock and class level lock, both are independent.

Q293. While a thread executing any synchronized method on the given object is it possible to execute remaining synchronized method of the same object simultaneously by any other thread?

Ans. No, it is no possible.

Q294. What is the difference between synchronized method and static synchronized method?

Ans. If a Thread wants to execute a synchronized method first it has to get the lock of the object. Once a Thread got the lock then it is allow to execute any synchronized method on that object.If a Thread wants to execute static synchronized method that Thread has to get class level lock once a Thread got class level lock then only it is allow to execute static synchronized method.

Q295. What is the advantage of synchronized block over synchronized method?

Ans. If very few lines of the code required synchronization then declaring entire method as the synchronized is not recommended. We have to declare those few lines of the code inside synchronized block. This approach reduces waiting time of the Thread and improves performance of the system.

Q296. What is synchronized statement?

Ans. The Statement which is inside the synchronized area (synchronized method or synchronized block) is called synchronized statement.

Q297. How we can declare synchronized block to get class level lock?

Ans. To get the class level lock we can declare synchronized block as follows:

```
synchronized(Display.class)
{
}
```

Q298. How two thread will communicate with each other?

Ans. Two Threads will communicate with each other by using wait(), notify(), notifyAll() methods.

Q299.wait(), notify(), notifyAll() method can available in which

class?
Ans. These methods are defined in Object class.

Q300.Why wait(), notify(), notifyAll() method defines in object class instead of Thread class?
Ans. These methods are defined in Object class but not in Thread because Threads are calling this method on the shared object.

Q301.If a waiting thread got notification then it will entered into which state?
Ans. It will entered into another waiting state to get lock.

Q302.In which method threads can release the lock?
Ans. Once a Thread calls wait() method it immediately releases the lock of that object and then entered into waiting state similarly after calling notify() method Thread releases the lock but may not immediately. Except these three methods(wait(), notify(), notifyAll()) method Thread never releases the lock anywhere else.

Q303. Explain wait(), notify(), notifyAll() method uses.
Ans. Two Threads will communicate with each other by using wait(), notify() or notifyAll() methods.
These methods are defined in Object class but not in Thread because Threads are calling this method.

Q304. What is the difference between notify() and notifyAll()?
Ans. To give notification to the single waiting Thread. We use notify() method and to give notification to all waiting thread we use notifyAll() method.

Q305. Once a Thread got the notification then which waiting thread will get chance?
Ans. It is depends on the Thread Scheduler.

Q306. How a thread can interrupt another thread?
Ans. A Thread can interrupt another Thread by using interrupt() method.

Q307. Which keyword causes DeadLock situation?
Ans. Synchronized keyword is the thing to causes of DeadLock. If we are not using properly synchronized keyword the program will entered into DeadLock situation.

Q308. How we can stop a thread explacitly?
Ans. Thread class defines stop() method by using this method we can stop a Thread. But it is deprecated. And hence not recommended to use.

Q309. Explain about suspend() and resume() method?
Ans. A Thread can suspend another Thread by using suspend() method. A Thread can resume a suspended Thread by using resume() method.

Q310.What is Starvation()? And Explain the difference between Deadlock and Starvation?
Ans. A long waiting Thread is said to be in starvation (because of least priority) but after certain time defiantly it will get the chance for execution. But in the case of Deadlock two Threads will wait for each other forever. It will never get the chance for execution.

Q311. What is race condition?
Ans. Multiple Threads are accessing simultaneously and causing data inconsistency problem is called race condition, we can resolve this by using synchronized keyword.

Q312. What is Daemon Thread? And give an example?
Ans. The Threads which are running in the background are called Daemon Thread.
Example: Garbage collector.

Q313. What is the purpose of a Daemon Thread?
Ans. The main purpose of Daemon Threads is to provide support for non-daemon Threads.

Q314. How we can check Daemon nature of a Thread?
Ans. We can check Daemon nature of a Thread by using isDaemon() method.

Q315. Is it possible to change a Daemon nature of a Thread?
Ans. Yes, we can change Daemon nature of a Thread by using setDaemon() method.

Q316. Is main thread is Daemon or non-daemon?
Ans. By default main thread is always non-daemon nature.

Q317. Once we created a new thread is it daemon or non-daemon.
Ans. Once we created a new Thread, The Daemon nature will be inheriting from parent to child. If the parent is Daemon the child is also Daemon and if the parent is non-daemon then child is also non-daemon.

Q318. After starting a thread is it possible to change Daemon nature?
Ans. We can change the Daemon nature before starting the Thread only. Once Thread started we are not allow to change Daemon nature otherwise we will get RuntimeException sying IllegalThreadStateException.

Q319. When the Daemon thread will be terminated?
Ans. Once last non-daemon Thread terminates automatically every Daemon Thread will be terminated.

Q320. What is green Thread?
Ans. A green thread refers to a mode of operation for the Java Virtual Machine (JVM) in which all code is executed in a single operating system thread. If the Java program has any concurrent
threads, the JVM manages multi-threading internally rather than using other operating system threads.
There is a significant processing overhead for the JVM to keep track of thread states and swap between them, so green thread mode has been deprecated and removed from more recent Java implementations.

Q321.Explain about Thread group?
Ans. Every Java thread is a member of a thread group. Thread groups provide a mechanism for
collecting multiple threads into a single object and manipulating those threads all at once, rather
than individually. For example, you can start or suspend all the threads within a group with a
single method call. Java thread groups are implemented by the ThreadGroup api class in the java.lang package.

Q322.What is the Thread Local?
Ans. It's a way for each thread in multi-threaded code to keep its own copy of an instance
variable. Generally, instance variable are shared between all threads that use an object; ThreadLocal is a way for each thread to keep its own copy of such a variable. The purpose might be that each thread keeps different data in that variable, or that the developer wants to avoid the overhead of synchronizing access to it.

Q323. What is inner class and when we should go for inner classes?
Some times we can declare a class inside another class such type of classes are called inner classes
Example
Class Car{

```
//more code here
Class Engine{
//more code here }
}
```

Without existing Car object there is no chance of existing Engine object, hence Engine class has declared inside Car class.

Q324.How many types of inner classes are present?

There are four types of inner classes are present
o Normal or regular inner class o Method local inner class o Anonymous inner class o Static nested class

Q325.What is method local inner class?

Sometimes we can declare a class inside a method such type of classes are called method local
inner classes
The main purpose of method local inner classes is to define method specific functionality The scope of method local inner classes is the scope of the method where it is declared. This is the mostly rarely used type of inner classes.
Example

```
class Test{
public void m1(){
class Inner {
public void sum(int I,int j){
System.out.println(i+J);
}//sum
}//inner
Inner i=new Inner();
i.sum(10,20);
//more code here
I.sum(100,303);
//more code here
i.sum(102,84);
}//m1()
Public static void main(){
New Test().m1();
}
}
```

Q326.What is anonymous inner class?

Some times we can declare a inner class without name such type of inner classes are called
Anonymous inner classes
Anonymous inner classes can be divided into 3 categories
§ Anonymous inner class that extends a class

§ Anonymous inner class that implements an interface
§ Anonymous inner class that defines inside a method argument
ANONYMOUS INNER CLASS THAT EXTENDS A CLASS Example

```
Class popcorn{
Public void taste(){
System.out.println("it is salty");
}
//more code here
}
Class Test{
Public static void main(String[] args)
{
Popcorn p=new Popcorn()
{ // here we are creating child class for popcorn
Public void taste(){
System.out.println("it is sweet");
}
};//here semicolon indicates we r creating child class object with parent
// class reference here child class dosent contain name
p.taste()// it is sweet
Popcorn p=new Popcorn();
p1.taste() //it is salty }
}
```

ANONYMOUS INNER CLASS THAT IMPLEMENTS AN INTERFACE example

```
class Test{
Public static void main(String[] args){
Runnable r=new Runnable(){
Public void run(){
for(int i=0;i<10;i++){
System.out.printin("child thread");
}
}
};
Thread t=new Thread(r);
t.start();
for(int i=0;i<10;i++){
System.out.printin("main thread"); }
}
}
```

Don't become fool that here we are creating object of interface Runnable.Here we are actually
creating an object of class that is implemented Runnable interface.

Q327.What is static nested calss?why the term nested instead of inner in static nested class?

Some times we can declare inner class with static modifier such type of inner class are called static
nested classes.the term nested instead of static because without existing outer class object inner
class object can exist.
Example
```
Class outer{
Static class Nested{
Public static void main(String[] args){
System.out.println("nested class main()"); }
}
Public static void main(String[] args){
System.out.println("outer class main()"); }
}
```
Java Outer
O/P
Outer class main() Java Outer$Nested
O/P
Nested class main()

Q328. Inside inner class is it possible to declare main()?
No it is not possible to declare main () inside inner class but in static nested class it is possible for
Example refer above code

Q329. What are limitations of object Arrays?
The main limitations of Object arrays are
* These are fixed in size ie once we created an array object there is no chance of increasing or decreasing size based on our requirement. Hence If we don't know size in advance , arrays are not recommended to use
* Arrays can hold only homogeneous elements.
* There is no underlying data structure for arrays and hence no readymade method support for arrays. Hence for every requirement programmer has to code explicitly
To over come these problems collections are recommended to use.

Q330. What is Collection API ?
It defines set of classes and interfaces which can be used for representing a group of objects as single entity

Q331. What is Collection framework?
It defines set of classes and inter faces which can be used for representing a group of objects as single entity

Q332. What is difference between Collections and Collection?
Collection is an interface which can be used for representing a group of individual objects as single entity and it acts as root interface of collection frame work.
Collections is an utility class to define several utility methods for Collection implemented class objects.

Q333. Explain about Collection interface?
* This interface can be used to represent a group of objects as a single entity. * It acts as root interface for entire collection framework.
* It defines the most commonly used methods which can be applicable for any collection implemented class object
Q339. Explain about List interface?
List interface is a child interface of Collection interface. This can be used to represent group of individual objects in as a single entity where
* Duplicates are allowed
* Insertion order is preserved

Q334. Explain about Set interface?
Set is a child interface of Collection interface. it can be used to represent a group of individual objects as a single entity where
* Duplicate objects are not allowed.
* Insertion order is not preserved

Q335. Explain about SortedSet interface?
it is child interface of Set interface. it can be used to represent a group of individual objects in to a single entity where
All the objects are arranged in some sorting order (Can be natural sorting order or customizede).
Duplicates are not allowed.

Q336. Explain about NavigableSet ?
It is child interface of SortedSet and provides several utility methods for navigation purposes
It doesn't allows duplicates
Insertion order is preserved
It is introduced in 1.6 version

Q337. Explain about Queue interface?

If we want to represent a group of individual objects prior to processing, then we should go for Queue interface. It is child interface of Collection interface.
It has introduced in 1.5 version.

Q338. Explain about Map interface?
Remember it is not a child Interface of Collection Interface and hence Map and Collection Interfaces doesn't have any relationship.
It can be used for representing a group of Objects as key, value pairs.
Both keys and values should be objects
Keys can t be duplicated but values can be duplicated. it has introduced in 1.2 version

Q339. Explain about SortedMap ?
If we want to represent a group of objects as key value pairs where all the entries are arranged according some sorting order of keys then we should go for SortedMap. It is child interface of Map.
It has introduced in 1.2 version

Q340. Explain about NavigableMap?
It is child interface of SortedMap and defines several method for navigation purpose It is introduced in 1.6 version

Q341. Explain about ArrayList class?
ArrayList is a Collection which can be used to represent a group of objects as a single entity.
it is a implemented class for List interface
Introduced in 1.2 version
The underlying data structure is resizable or growable array. Insertion order is preserved
Duplicates are allowed
Heterogeneous objects are allowed
null insertion is possible
This class implements RandomAccess , Serializable , Cloneable interfaces
Best choice for retrieval purpose and worst if our frequent operation is insertion or deletion in the middle

Q342. What is RandomAccess Interface?
If a collection class implements RandomAccess interface then we can access any of its element with the same speed.
RandomAccess interface is marker interface and it dosent contains any methods. ArrayList and vector classes implements this interface.

Q343. Explain about LinkedList class?
LinkedList is a Collection implemented class which can be used for representing a group of objects as a single entity.
LinkedList is the implemetation class for List interface Introduced in 1.2

version
Underlying data Structure is DoubleLinkedList Allows duplicates
Insertion order is preserved
Allows heterogeneous objects
null insertion is possible
LinkedList class implements Seriallizable and Cloneable interface but not RandomAccess interface
Best choice if frequent operation is insertion or deletion an objects in middle but worst choice if frequent operation is retrieval.

Q344. Explain about Vector class?
Vector is a legacy collection class which can be used to represent a group of objects.
Introduced in 1.0 version. it is legacy class
The underlying data structure is resizable or growable array. Insertion order is preserved
Duplicates are allowed
Heterogeneous objects are allowed
It is a implemented class for List interface null insertion is possible
Vector class implements RandomAccess ,Serializable,Cloneable interfaces
Best Choice if frequent operation is retrieval and worst choice if frequent operation is insertion or deletion in the middle.
All methods present in Vector class are synchronized hence Vector class object is thread safe.

Q345. How we can get synchronized version of ArrayList?
Collections class contains synchronizedList() method for this
Public static List synchronizedList(List l)
EX
ArrayList l= new ArrayList();
List l2=Collections.synchronizedList(l);
Similarly we can get synchronized versions of Set and Map objects by the following methods. Public static List synchronizedSet(Set s)
Public static List synchronizedMap(Map m)

Q346. What is difference between size and capacity of a Collection Object?
size means number of objects present where as capacity means no of objects it can accommodate.

Q347. What are legacy classes and interfaces present in Collections framework ?
Enumeration ---Interface
Dictonary ------Abstract class

Hashtable -----Concrete class
Properties -----Concrete class
Vector -----Concrete class Stack -----Concrete class

Q348. What are limitations of Enumeration?
While iterating the elements we are not allowed to perform removal operation It is applicable only for legacy classes and it is not a universal cursor. It can retrieve the elements only in forward direction.

Q349. What is difference between enum and Enumeration?
An enum can be used to define a group of named constants .It has introduced in 1.5 version Ex
Class Beer{
KO,KF,RC,FO
}
Enumeration is cursor to retrieve Objects one by one from Collection objects.

Q350. What is difference between Iterator and ListIterator?
o ListIterator is the child interface of the Iterator
o Iterator is the single direction cursor where as ListIterator is bidirectional cursor. o While iterating the elements by Iterator we can perform only read and remove
operations. But by using ListIterator we can perform read,removal, replace and addition of new objects also.
o Iterator is applicable for every Collecton implemented class object but ListIterator is applicable only for List implemented class objects.
o Iterator can be get by using iterator() of Collection interface where as ListIterator
can be get by using listIterator() method of List interface
o both are introduced in 1.2 version

Q351. What is relation between ListIterator and Iterator?
ListIterator is child interface of Iterator

Q352. Explain about HashSet class?
The underlying data structure is Hashtable null values are accepted duplicates are not allowed
insertion order is based on hashcode of the object hence insertion order is not preserved best suitable if frequent operation is search operations
HashSet class implements Serializable and Cloneable it is implementation class for Set interface
heterogeneous objects are allowed
it is introduced in 1.2 version

Q353. If we are trying to insert duplicate values in Set what will happen?
If we are trying to insert duplicate objects to the HashSet , we wont get any compile time or run time errors just the add(Object o) returns false and it doesn't add that object.

Q354. What is LinkedHashSet?
It is the child class of HashSet. The main difference between HashSet and LinkedHashSet is: In the case of HashSet insertion order is not preserved , but in the case of LinkedHashSet insertion will be preserved.

Q355. What are major enhancements in 1.4 version of collection frame work?
LinkedHashSet
LinkedHashMap
IdentityHashMap

Q356. Explain about TreeSet?
It is Collection object which can be used to represent a group of objects according to some sorting order.
The underlying datastructure is Balanced tree Duplicates are not allowed
All objects are stored according to some sorting order hence insertion order is not preserved
Heterogeneous objects are not allowed violation leads to ClassCastException
For an Empty TreeSet as firs element null value can be inserted but after inserting that first value if we are trying to insert any other objects then we will get NullPointerException
For an non empty TreeSet if we are trying to inser null value at run time u will get NullPointerException

Q357. What is Comparable interface?
This interface can be used for defining natural sorting order of the objects. It is present in java.lang package
It contains a method public int compareTo(Object obj1)

Q358. What is Comparator interface?
This interface can be used for implementing customized sorting order. It is present in java.util package
It contains two methods
o public int compare(Object ,Object) o public boolean equals(Object)

Q359. What is Entry interface?
It is inner interface of Map.
In the Map each key value pair is considered as Entry object.
interface Map{
//more code here
interface Entry{
Object getKey()
Object getValue()
Object setValue(Object new)
}
}

Q360. Explain about HashMap?
It is a Map Object which can be used used to represent a group of objects as key-value pairs.
The underlying data structure is Hashtable
Duplicaes keys are not allowed duplicate values are allowed
Insertion order is not preserved because insertion is based on hashcode of keys. Heterogeneous objects are allowed for both keys and values
null key is allowed only once
null values are allowed multiple times
Introduced in 1.2 version

Q361. Explain about LinkedHashMap?
It is child class of HashMap. It is exactly same as HashMap except the following difference. In the case of HashMap the insertion order is not preserved but in the case of LinkedHashMap insertion order is preserved.
Introduced in 1.4 version

Q362. What is IdentityHashMap?
It is exactly same as HashMap except the following difference.
In the HashMap JVM uses equals() method to identify duplicate keys but in the case of IdentityHashMap JVM uses == operator for this.

Q363. What is difference between HashMap and IdentityHashMap?
Refer Q377 for the answer.

Q364. What is WeakHashMap?
It is exactly same as HashMap except the following difference.
In case of HashMap an Object is not eligible for garbage collection if it is associated with
HashMap even though it dosent have any external references. ie HashMap

dominates garbage collector.
But in case of WeakHashMap , if an Object is not having any external references then it is always eligible for garabage collectoion even though it is associated with weakHashMap. ie garbage collector dominates WeakHashMap

Q365. What is difference between HashMap and WeakHashMap?
Refer Q379 for the answer.

Q366. What is TreeMap?
TreeMap can be used to store a group of objects as key-value pairs where all the entries are arranged according to some sorting order of keys.
The underlying data structure is RED-BLACK Tree
Duplicates keys are not allowed but values can be duplicated.
Insertion order is not preserved because insertion is based on some sorting order
If we are depending on Natural sorting order then keys should be homogeneous(violation leads to ClassCastException) but values need not homogeneous
In case of customized sorting order we can insert heterogeneous keys and values
For empty TreeMap as first entry with null values are allowed but after inserting that entry if we are trying to insert any other entry we will get NullPointerException For non empty TreeMap if we are trying to insert null keys we will get
NullPointerException
There are no restrictions for null values.

Q367. What is Hashtable?
Hashtable is a legacy Map and can be used to store objects as key value pairs.
The underlying data sturucture is Hashtabe
Duplicates keys are not allowed but duplicate values are allowed null insertion is not possible for both keys and values
all methods are synchronized
insertion order is not preserved because it is based on hashcode of keys
heterogeneous Objects are allowed for both keys and values
introduced in 1.0 version it is legacy class

Q368. What is PriorityQueue?
It represents a data structure to hold group of individual objects prior to processing based on
some priority .it can be natural sorting order and it can be customized sorting order described by Comparator.
It is the implementation class of Queue interface.

Insertion order is not preserved because here insertion is done based on some sorting order
Duplicates are not allowed
null insertion is not possible even as first element also
If we are depending on natural sorting order Objects should be homogeneous violation leads to ClassCastException
If we are depending on customized sorting order Objects can be heterogeneous also.

Q369. What is Arrays class?
It is utility class for arrays.
It defines several utility methods for arrays like sorting an array or searching an element in array
present in java.util package

Q370. We are planning to do an indexed search in a list of objects. Which of the two Java collections should you use: ArrayList or LinkedList?
ArrayList

Q371. Why ArrayList is faster than Vector?
All methods present in the Vector are synchronized and hence any method can be executed by only one thread at a time. It slows down the execution.
But in ArrayList, no method is synchronized and hence multiple thread are allowed execute simultaneously which speed up the execution.