

## **JAVA INTERVIEW QUESTIONS AND ANSWERS**

### **1. What does a well-written OO program look like?**

A well-written OO program exhibits recurring structures that promote abstraction, flexibility, modularity and elegance.

### **2. Can you have virtual functions in Java?**

Yes, all functions in Java are virtual by default. This is actually a pseudo trick question because the word "virtual" is not part of the naming convention in Java (as it is in C++, C-sharp and VB.NET), so this would be a foreign concept for someone who has only coded in Java. Virtual functions or virtual methods are functions or methods that will be redefined in derived classes.

### **3. Jack developed a program by using a Map container to hold key/value pairs. He wanted to make a change to the map. He decided to make a clone of the map in order to save the original data on side. What do you think of it? ?**

If Jack made a clone of the map, any changes to the clone or the original map would be seen on both maps, because the clone of Map is a shallow copy. So Jack made a wrong decision.

### **4. What is more advisable to create a thread, by implementing a Runnable interface or by extending Thread class?**

Strategically speaking, threads created by implementing Runnable interface are more advisable. If you create a thread by extending a thread class, you cannot extend any other class. If you create a thread by implementing Runnable interface, you save a space for your class to extend another class now or in future.

### **5. What is NullPointerException and how to handle it?**

When an object is not initialized, the default value is null. When the following things happen, the NullPointerException is thrown: --Calling the instance method of a null object. --Accessing or modifying the field of a null object. --Taking the length of a null as if it were an array. --Accessing or modifying the slots of null as if it were an array. - -Throwing null as if it were a Throwable value. The NullPointerException is a runtime exception. The best practice is to catch such exception even if it is not required by language design.

### **6. An application needs to load a library before it starts to run, how to code??**

One option is to use a static block to load a library before anything is called. For example, class Test { static { System.loadLibrary("path-to-library-file"); } .... } When you call new Test(), the static block will be called first before any initialization happens. Note that the static block position may matter.

### **7. What is a platform?**

A platform is the hardware or software environment in which a program runs. Most platforms can be described as a combination of the operating system and hardware, like Windows 2000 and XP, Linux, Solaris, and MacOS

### **8. What is the main difference between Java platform and other platform§**

The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has three elements:

Java programming language

The Java Virtual Machine (Java VM)

The Java Application Programming Interface (Java API)

### **9. What is the Java Virtual Machine?**

The Java Virtual Machine is a software that can be ported onto various hardware-based platforms.

### **10. What is the Java API?**

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

### **11. What is the package?**

The package is a Java namespace or part of Java libraries. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

### **12. What is native code?**

The native code is code that after you compile it, the compiled code runs on a specific hardware platform.

### **13. Is Java code slower than native code?**

Not really. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

### **14. Can main() method be overloaded?**

Yes. the main() method is a special method for a program entry. You can overload main() method in any ways. But if you change the signature of the main method, the entry point for the program will be gone.

### **15. What is the serialization?**

The serialization is a kind of mechanism that makes a class or a bean persistence by having its properties or fields and state information saved and restored to and from storage.

### **16. How to make a class or a bean serializable?**

By implementing either the java.io.Serializable interface, or the java.io.Externalizable interface. As long as one class in a class's inheritance hierarchy implements Serializable or Externalizable, that class is serializable.

### **17. How many methods in the Serializable interface?**

There is no method in the Serializable interface. The Serializable interface acts as a marker, telling the object serialization tools that your class is serializable.

### **18. How many methods in the Externalizable interface?**

There are two methods in the Externalizable interface. You have to implement these two methods in order to make your class externalizable. These two methods are readExternal() and writeExternal().

### **19. What is the difference between Serializable and Externalizable interface?**

When you use Serializable interface, your class is serialized automatically by default. But you can override writeObject() and readObject() two methods to control more complex object serialization process. When you use Externalizable interface, you have a complete control over your class's serialization process.

**20. What is a transient variable?**

A transient variable is a variable that may not be serialized. If you don't want some field to be serialized, you can mark that field transient or static.

**21. Which containers use a border layout as their default layout?**

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

**22. 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.

**23. 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 causes dirty data and leads to significant errors.

**24. What are synchronized methods and synchronized statements?**

Synchronized methods are methods that are used to control access to a method or 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.

**25. What are three ways in which a thread can enter the waiting state?**

A thread can enter the waiting state by invoking its sleep() method, by blocking on IO, 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.

**26. 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.

**27. 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

**28. 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.

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

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

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

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

**31. What is thread?**

A thread is an independent path of execution in a system.

**32. What is multi-threading?**

Multi-threading means various threads that run in a system.

**33. How does multi threading 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.

**34. How to create a thread in a program**

You have two ways to do so. First, making your class "extends" Thread class. Second, making your class "implements" Runnable interface. Put jobs in a run() method and call start() method to start the thread.

**35. Can Java object be locked down for exclusive use by a given thread?**

Yes. You can lock an object by putting it in a "synchronized" block. The locked object is inaccessible to any thread other than the one that explicitly claimed it.

**36. Can each Java object keep track of all the threads that want to exclusively access to it?**

Yes. Use Thread.currentThread() method to track the accessing thread.

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

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

**38. What invokes a thread's run() method?**

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

**39. What is the purpose of the wait(), notify(), and notifyAll() methods?**

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

**40. What are the high level thread states?**

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

**41. 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.

**42. 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.

**43. What is the difference between Process and Thread?**

A process can contain multiple threads. In most multithreading operating systems, a process gets its own memory address space; a thread doesn't. Threads typically share the heap belonging to their parent process. For instance, a JVM runs in a single process in the host O/S. Threads in the JVM share the heap belonging to that process; that's why several threads may access the same object. Typically, even though they share a common heap, threads have their own stack space. This is how one thread's invocation of a method is kept separate from another's. This is all a gross oversimplification, but it's accurate enough at a high level. Lots of details differ between operating systems.

| Process            | Thread                                                    |
|--------------------|-----------------------------------------------------------|
| A program          | similar to a sequential program                           |
| run on its own     | Cannot run on its own                                     |
| Unit of allocation | vs.                                                       |
| Unit of execution  | Have its own memory space                                 |
| vs.                | Share with others                                         |
| Each process       | has one or more threads                                   |
| vs.                | Each thread belongs to one process                        |
| Expensive,         | need to context switch                                    |
| vs.                | Cheap, can use process memory and may not need to context |

switch More secure. One process cannot corrupt another process vs. Less secure. A thread can write the memory used by another thread

**44. What is the Collections API?**

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

**45. What is the List interface?**

The List interface provides support for ordered collections of objects.

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

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

**47. What modifiers may be used with an inner class that is a member of an outer class?**

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

**48. If a method is declared as protected, where may the method be accessed?**

A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

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

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

**50. 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.

**51. Is sizeof a keyword?**

The sizeof operator is not a keyword in Java.

**52. What are wrapped classes?**

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

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

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

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

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

**55. Name Component subclasses that support painting?**

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

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

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

**57. How can you write a loop indefinitely?**

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

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

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

**59. 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.

**60. Which class is the superclass for every class.**

Object.

**61. How many methods in Object class?**

This question is not asked to test your memory. It tests you how well you know Java. Ten in total.

clone()

equals() & hashCode()

getClass()

finalize()

wait() & notify()

toString()

**62. How does Java handle integer overflows and underflows?**

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

**63. What is the numeric promotion?**

Numeric promotion is used with both unary and binary bitwise operators. This means that byte, char, and short values are converted to int values before a bitwise operator is applied.

If a binary bitwise operator has one long operand, the other operand is converted to a long value.

The type of the result of a bitwise operation is the type to which the operands have been promoted. For example:

short a = 5;

byte b = 10;

long c = 15;

The type of the result of (a+b) is int, not short or byte. The type of the result of (a+c) or (b+c) is long.

**64. Is the numeric promotion available in other platform?**

Yes. Because Java is implemented using a platform-independent virtual machine, bitwise operations always yield the same result, even when run on machines that use radically different CPUs.

**65. What is the difference between the Boolean & operator and the && operator?**

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

Operator & has no chance to skip both sides evaluation and && operator does. If asked why, give details as above.

**65. When is the ArithmeticException throwQuestion: What is the GregorianCalendar class?**

The GregorianCalendar provides support for traditional Western calendars.

**66. What is the SimpleTimeZone class?**

The SimpleTimeZone class provides support for a Gregorian calendar.

**67. Which Container method is used to cause a container to be laid out and redisplayed?**

validate()

**68. What is the Properties class?**

The properties class is a subclass of Hashtable that can be read from or written to a stream. It also provides the capability to specify a set of default values to be used.

**69. What is the purpose of the Runtime class?**

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

**70. What is the purpose of the System class?**

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

**71. What is the purpose of the finally clause of a try-catch-finally statement?**

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

**72. 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.

**73. What is an abstract method?**

An abstract method is a method whose implementation is deferred to a subclass. Or, a method that has no implementation.

**74. What is the difference between interface and abstract class?**

interface contains methods that must be abstract; abstract class may contain concrete methods. interface contains variables that must be static and final; abstract class may contain non-final and final variables. members in an interface are public by default, abstract class may contain non-public members. interface is used to "implements"; whereas abstract class is used to "extends". interface can be used to achieve multiple inheritance; abstract class can be used as a single inheritance. interface can "extends" another interface, abstract class can "extends" another class and "implements" multiple interfaces. interface is absolutely abstract; abstract class can be invoked if a main() exists. interface is more flexible than abstract class because one class can only "extends" one super class, but "implements" multiple interfaces. If given a choice, use interface instead of abstract class.

**75. What is a static method?**

A static method is a method that belongs to the class rather than any object of the class and doesn't apply to an object or even require that any objects of the class have been instantiated.

**76. What is a protected method?**

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

**77. What is the difference between a static and a non-static inner class?**

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

**78. What is an object's lock and which object's have locks?**

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

**79. When can an object reference be cast to an interface reference?**

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

**80. What is the difference between a Window and a Frame?**

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

**81. What is the difference between a Window and a Frame?**

Heavy weight components like Abstract Window Toolkit (AWT), depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button. In this relationship, the Motif button is called the peer to the java.awt.Button. If you create two Buttons, two peers and hence two Motif Buttons are also created. The Java platform communicates with the Motif Buttons using the Java Native Interface. For each and every component added to the application, there is an additional overhead tied to the local windowing system, which is why these components are called heavy weight.

**82. Which package has light weight components?**

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

**83. What are peerless components?**

The peerless components are called light weight components.

**84. What is the difference between the Font and FontMetrics classes?**

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

**85. 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.

**86. 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.

**87. What is the difference between throw and throws keywords?**

The throw keyword denotes a statement that causes an exception to be initiated. It takes the Exception object to be thrown as argument. The exception will be caught by an immediately encompassing try-catch construction or propagated further up the calling hierarchy.

The throws keyword is a modifier of a method that designates that exceptions may

come out of the method, either by virtue of the method throwing the exception itself or because it fails to catch such exceptions that a method it calls may throw.

**88. 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 or friendly access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

**89. What is the Map interface?**

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

**90. Does a class inherit the constructors of its superclass?**

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

**91. Name primitive Java types.**

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

**92. Which class should you use to obtain design information about an object?**

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

**92. 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.

**93. 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.

**94. What advantage do Java's layout managers provide over traditional windowing systems?**

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

**95. What are the problems faced by Java programmers who don't use layout managers?**

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

**96. What is the difference between static and non-static variables?**

A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

**97. 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.

**98. 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.

**99. What restrictions are placed on method overloading?**

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

**100. What restrictions are placed on method overriding?**

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

**101. 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.

**102. Name Container classes.**

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

**103. What class allows you to read objects directly from a stream?**

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

**104. 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.

**105. How is it possible for two String objects with identical values not to be equal under the == operator? How are this() and super() used with constructors?**

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

**106. What is an IO filter?**

An IO 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.

**107. 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.

**108. What is the List interface?**

The List interface provides support for ordered collections of objects.

**109. What is the purpose of the enableEvents() method?**

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

**110. What is the difference between the File and RandomAccessFile classes?**

The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

**111. What interface must an object implement before it can be written to a stream as an object?**

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

**112. 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.

**113. What is the difference between a Scrollbar and a ScrollPane?**

A Scrollbar is a Component, but not a Container. A ScrollPane is a Container. A ScrollPane handles its own events and performs its own scrolling.

**114. 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.

**115. 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.

**116. What is Serialization and deserialization?**

Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

**117. What is tunnelling?**

Tunnelling is a route to somewhere. For example, RMI tunnelling is a way to make RMI application get through firewall. In CS world, tunnelling means a way to transfer data.

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

If an exception occurs and there is a return statement in catch block, the finally block is still executed. The finally block will not be executed when the System.exit(1) statement is executed earlier or the system shut down earlier or the memory is used up earlier before the thread goes to finally block.

**119. How do you restrict a user to cut and paste from the html page?**

Using Servlet or client side scripts to lock keyboard keys. It is one of solutions.

**120. Is Java a super set of JavaScript?**

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

**121. What is a Container in a GUI?**

A Container contains and arranges other components (including other containers) through the use of layout managers, which use specific layout policies to determine where components should go as a function of the size of the container.

**122. How the object oriented approach helps us keep complexity of software development under control?**

We can discuss such issue from the following aspects:

1. Objects allow procedures to be encapsulated with their data to reduce potential interference.
2. Inheritance allows well-tested procedures to be reused and enables changes to make once and have effect in all relevant places.
3. The well-defined separations of interface and implementation allow constraints to be imposed on inheriting classes while still allowing the flexibility of overriding and overloading.

### **123. What is polymorphism?**

Polymorphism means "having many forms". It allows methods (may be variables) to be written that needn't be concerned about the specifics of the objects they will be applied to. That is, the method can be specified at a higher level of abstraction and can be counted on to work even on objects of un-conceived classes.

### **124. What is design by contract?**

The design by contract specifies the obligations of a method to any other methods that may use its services and also theirs to it. For example, the preconditions specify what the method required to be true when the method is called. Hence making sure that preconditions are. Similarly, postconditions specify what must be true when the method is finished, thus the called method has the responsibility of satisfying the post conditions.

In Java, the exception handling facilities support the use of design by contract, especially in the case of checked exceptions. The assert keyword can be used to make such contracts.

### **125. What are use cases?**

A use case describes a situation that a program might encounter and what behavior the program should exhibit in that circumstance. It is part of the analysis of a program. The collection of use cases should, ideally, anticipate all the standard circumstances and many of the extraordinary circumstances possible so that the program will be robust.

### **126. What is scalability and performance?**

Performance is a measure of "how fast can you perform this task." and scalability describes how an application behaves as its workload and available computing resources increase.

### **127. What is the benefit of subclass?**

Generally: The sub class inherits all the public methods and the implementation.

The sub class inherits all the protected methods and their implementation.

The sub class inherits all the default(non-access modifier) methods and their implementation.

The sub class also inherits all the public, protected and default member variables from the super class.

The constructors are not part of this inheritance model.

### **128. How to add menushortcut to menu item?**

If you have a button instance called aboutButton, you may add menu short cut by calling aboutButton.setMnemonic('A'), so the user may be able to use Alt+A to click the button.

**129. Where and how can you use a private constructor.**

Private constructor can be used if you do not want any other class to instantiate it by using new. The instantiation is done from a static public method, which is used when dealing with the factory method pattern.

**130. In System.out.println(), what is System, out and println pls explain?**

System is a predefined final class, out is a PrintStream object acting as a field member and println is a built-in overloaded method in the out object.

**131. What is meant by "Abstract Interface"?**

First, an interface is abstract. That means you cannot have any implementation in an interface. All the methods declared in an interface are abstract methods or signatures of the methods.

**132. Can you make an instance of an abstract class? For example - java.util.Calendar is an abstract class with a method getInstance() which returns an instance of the Calendar class.**

No! You cannot make an instance of an abstract class. An abstract class has to be sub-classed. If you have an abstract class and you want to use a method which has been implemented, you may need to subclass that abstract class, instantiate your subclass and then call that method.

**133. What is the output of x > y? a:b = p\*q when x=1,y=2,p=3,q=4?**

When this kind of question has been asked, find the problems you think is necessary to ask back before you give an answer. Ask if variables a and b have been declared or initialized. If the answer is yes. You can say that the syntax is wrong. If the statement is rewritten as: x< returned. is a variable and true return statement y="2;" x the than less 1 because be would value a:(b="p\*q");">

**134. What is the difference between Swing and AWT components?**

AWT components are heavy-weight, whereas Swing components are lightweight. Heavy weight components depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button.

**135. Why Java does not support pointers?**

Because pointers are unsafe. Java uses reference types to hide pointers and programmers feel easier to deal with reference types without pointers. This is why Java and C-sharp shine.

**136. Parsers? DOM vs SAX parser**

Parsers are fundamental xml components, a bridge between XML documents and applications that process that XML. The parser is responsible for handling xml syntax, checking the contents of the document against constraints established in a DTD or Schema.

DOM

1. Tree of nodes
2. Memory: Occupies more memory, preferred for small XML documents
3. Slower at runtime
4. Stored as objects
5. Programmatically easy
6. Ease of navigation

SAX

1. Sequence of events

2. Doesn't use any memory preferred for large documents
3. Faster at runtime
4. Objects are to be created
5. Need to write code for creating objects
6. Backward navigation is not possible as it sequentially processes the document

**137. Can you declare a class as private?**

Yes, we can declare a private class as an inner class. For example,

```
class MyPrivate {
    private static class MyKey {
        String key = "12345";
    }
    public static void main(String[] args) {
        System.out.println(new MyKey().key); //prints 12345
    }
}
```

**138. What is the difference between shallow copy and deep copy?**

Shallow copy shares the same reference with the original object like cloning, whereas the deep copy get a duplicate instance of the original object. If the shallow copy has been changed, the original object will be reflected and vice versa.

**139. Can one create a method which gets a String and modifies it?**

No. In Java, Strings are constant or immutable; their values cannot be changed after they are created, but they can be shared. Once you change a string, you actually create a new object. For example:

```
String s = "abc"; //create a new String object representing "abc"
s = s.toUpperCase(); //create another object representing "ABC"
```

**140. Why is multiple inheritance not possible in Java?**

It depends on how you understand "inheritance". Java can only "extends" one super class, but can "implements" many interfaces; that doesn't mean the multiple inheritance is not possible. You may use interfaces to make inheritance work for you. Or you may need to work around. For example, if you cannot get a feature from a class because your class has a super class already, you may get that class's feature by declaring it as a member field or getting an instance of that class. So the answer is that multiple inheritance in Java is possible.

**141. Can Java code be compiled to machine dependent executable file?**

Yes. There are many tools out there. If you did so, the generated exe file would be run in the specific platform, not cross-platform.

**142. What is the relationship between synchronized and volatile keyword?**

The JVM is guaranteed to treat reads and writes of data of 32 bits or less as atomic. (Some JVM might treat reads and writes of data of 64 bits or less as atomic in future) For long or double variable, programmers should take care in multi-threading environment. Either put these variables in a synchronized method or block, or declare them volatile.

**143. This class (IncrementImpl) will be used by various threads concurrently; can you see the inherent flaw(s)? How would you improve it?**

```
public class IncrementImpl {
```

```
private static int counter = 0;
public synchronized void increment() {
    counter++;
}
public int getCounter() {
    return counter;
}
}
```

The counter is static variable which is shared by multiple instances of this class. The increment() method is synchronized, but the getCounter() should be synchronized too. Otherwise the Java run-time system will not guarantee the data integrity and the race conditions will occur. The famous producer/consumer example listed at Sun's thread tutorial site will tell more.

one of solutions

```
public class IncrementImpl {
    private static int counter = 0;
    public synchronized void increment() {
        counter++;
    }
    public synchronized int getCounter() {
        return counter;
    }
}
```

#### **144. What are the drawbacks of inheritance?**

Since inheritance inherits everything from the super class and interface, it may make the subclass too clustering and sometimes error-prone when dynamic overriding or dynamic overloading in some situation. In addition, the inheritance may make peers hardly understand your code if they don't know how your super-class acts and add learning curve to the process of development.

Usually, when you want to use a functionality of a class, you may use subclass to inherit such function or use an instance of this class in your class. Which is better, depends on your specification.

#### **145. Is there any other way that you can achieve inheritance in Java?**

There are a couple of ways. As you know, the straight way is to "extends" and/or "implements". The other way is to get an instance of the class to achieve the inheritance. That means to make the supposed-super-class be a field member. When you use an instance of the class, actually you get every function available from this class, but you may lose the dynamic features of OOP

#### **146. Two methods have key words static synchronized and synchronized separately. What is the difference between them?**

Both are synchronized methods. One is instance method, the other is class method. Method with static modifier is a class method. That means the method belongs to class itself and can be accessed directly with class name and is also called Singleton design. The method without static modifier is an instance method. That means the instance method belongs to its object. Every instance of the class gets its own copy of its instance method.

When synchronized is used with a static method, a lock for the entire class is obtained. When synchronized is used with a non-static method, a lock for the particular object (that means instance) of the class is obtained.

Since both methods are synchronized methods, you are not asked to explain what is a synchronized method. You are asked to tell the difference between instance and class method. Of course, your explanation to how synchronized keyword works doesn't hurt. And you may use this opportunity to show your knowledge scope.

**147. How do you create a read-only collection?**

The Collections class has six methods to help out here:

1. `unmodifiableCollection(Collection c)`
2. `unmodifiableList(List list)`
3. `unmodifiableMap(Map m)`
4. `unmodifiableSet(Set s)`
5. `unmodifiableSortedMap(SortedMap m)`
6. `unmodifiableSortedSet(SortedSet s)`

If you get an Iterator from one of these unmodifiable collections, when you call `remove()`, it will throw an `UnsupportedOperationException`.

**148. Can a private method of a superclass be declared within a subclass?**

Sure. A private field or method or inner class belongs to its declared class and hides from its subclasses. There is no way for private stuff to have a runtime overloading or overriding (polymorphism) features.

**149. Why Java does not support multiple inheritance ?**

This is a classic question. Yes or No depends on how you look at Java. If you focus on the syntax of "extends" and compare with C++, you may answer 'No' and give explanation to support you. Or you may answer 'Yes'. Recommend you to say 'Yes'. Java DOES support multiple inheritance via interface implementation. Some people may not think in this way. Give explanation to support your point.

**150. What is the difference between final, finally and finalize?**

Short answer:

`final` - declares constant

`finally` - relates with exception handling

`finalize`- helps in garbage collection

If asked to give details, explain:

`final` field, `final` method, `final` class

`try/finally`, `try/catch/finally`

`protected void finalize()` in `Object` class

**151. What kind of security tools are available in J2SE 5.0**

There are three tools that can be used to protect application working within the scope of security policies set at remote sites.

`keytool` -- used to manage keystores and certificates.

`jarsigner` -- used to generate and verify JAR signatures.

`policytool` -- used for managing policy files.

There are three tools that help obtain, list and manage Kerberos tickets.

`kinit` -- used to obtain Kerberos V5 tickets.

`tklist` -- used to list entries in credential cache and key tab.

`ktab` -- used to help manage entries in the key table.

**152. How to make an array copy from System?**

There is a method called `arraycopy` in the `System` class. You can do it:

```
System.arraycopy(sourceArray, srcOffset, destinationArray, destOffset,
numOfElements2Copy);
```

When you use this method, the destinationArray will be filled with the elements of sourceArray at the length specified.

**153. Can we use System.arraycopy() method to copy the same array?**

Yes, you can. The source and destination arrays can be the same if you want to copy a subset of the array to another area within that array.

**154. What is shallow copy or shallow clone in array cloning?**

Cloning an array involves creating a new array of the same size and type and copying all the old elements into the new array. But such copy is called shallow copy or shallow clone because any changes to the object would be reflected in both arrays.

**155. When is the ArrayStoreException thrown?**

When copying elements between different arrays, if the source or destination arguments are not arrays or their types are not compatible, an ArrayStoreException will be thrown.

**156. How to check two arrays to see if contents have the same types and contain the same elements?**

One of options is to use the equals() method of Arrays class.

```
Arrays.equals(a, b);
```

If the array types are different, a compile-time error will happen.

**157. Do not use the String contatenation operator in lengthy loops or other places where performance could suffer. Is that true?**

Yes.

**158. The superclass constructor runs before the subclass constructor. The subclass's version of the overridable method will be invoked before the subclass's constructor has been invoked. If the subclass's overridable method depends on the proper initialization of the subclass (through the subclass constructor), the method will most likely fail. Is that true?**

Yes. It is true

**159. Why are the interfaces more flexible than abstract classes?**

--An interface-defined type can be implemented by any class in a class hierarchy and can be extended by another interface. In contrast, an abstract-class-defined type can be implemented only by classes that subclass the abstract class.

--An interface-defined type can be used well in polymorphism. The so-called interface type vs. implementation types.

--Abstract classes evolve more easily than interfaces. If you add a new concrete method to an abstract class, the hierarchy system is still working. If you add a method to an interface, the classes that rely on the interface will break when recompiled.

--Generally, use interfaces for flexibility; use abstract classes for ease of evolution (like expanding class functionality).

**160. What are new language features in J2SE 5.0?**

Generally:

1. generics
2. static imports
3. annotations
4. typesafe enums
5. enhanced for loop

- 6. autoboxing/unboxing
- 7. varargs
- 8. covariant return types

**161. What is covariant return type?**

A covariant return type lets you override a superclass method with a return type that subtypes the superclass method's return type. So we can use covariant return types to minimize upcasting and downcasting.

```
class Parent {
    Parent foo () {
        System.out.println ("Parent foo() called");
        return this;
    }
}

class Child extends Parent {
    Child foo () {
        System.out.println ("Child foo() called");
        return this;
    }
}

class Covariant {

    public static void main(String[] args) {
        Child c = new Child();
        Child c2 = c.foo(); // c2 is Child
        Parent c3 = c.foo(); // c3 points to Child
    }
}
```

**162. What is the result of the following statement?**

```
int i = 1, float f = 2.0f;
i += f; //ok, the cast done automatically by the compiler
i = i + f; //error
```

The compound assignment operators automatically include cast operations in their behaviors.

**163. The following statement prints true or false, why?**

```
byte[] a = { 1, 2, 3 };
byte[] b = (byte[]) a.clone();
System.out.println(a == b);
```

The false will be printed out. Because the two arrays have distinctive memory addresses. Starting in Java 1.2, we can use `java.util.Arrays.equals(a, b)` to compare whether two arrays have the same contents.

**164. Why do we need to use `getSystemResource()` and `getSystemResources()` method to load resources?**

Because we want to look for resources strictly from the system classpath, These methods use the system `ClassLoader` to locate resources, which gives you stricter control of the resources used by the application.

**165.ArithmeticException?**

The ArithmeticException is thrown when integer is divided by zero or taking the remainder of a number by zero. It is never thrown in floating-point operations.

**166. What is a transient variable?**

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

**167. Which containers use a border Layout as their default layout?**

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

**168. 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.

**169. 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.

**170. 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.

**171. 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.

**172. 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

**173. Is null a keyword?**

The null value is not a keyword.

**174. 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.

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

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

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

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

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

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

**178. What is the Collections API?**

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

**179. which characters may be used as the second character of an identifier, but not as the first character of an identifier?**

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

**180. What is the List interface?**

The List interface provides support for ordered collections of objects.

**181. How does Java handle integer overflows and underflows?**

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

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

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

**183. What modifiers may be used with an inner class that is a member of an outer class?**

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

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

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

**185. 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.

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

setBounds()

**187. 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.

**188. 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.

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

The EventObject class and the EventListener interface support event processing.

**190. Is sizeof a keyword?**

The sizeof operator is not a keyword.

**191. What are wrapper classes?**

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

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

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

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

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

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

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

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

Panel

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

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

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

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

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

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

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

Window.

**200. What is clipping?**

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

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

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

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

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

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

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

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

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

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

The default value of a String type is null.

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

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

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

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

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

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

**209. 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.

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

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

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

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

**212. What is the range of the short type?**

The range of the short type is  $-(2^{15})$  to  $2^{15} - 1$ .

**213. What is the range of the char type?**

The range of the char type is 0 to  $2^{16} - 1$ .

**214. In which package are most of the AWT events that support the event-delegation model defined?**

Most of the AWT-related events of the event-delegation model are defined in the java.awt.event package. The AWTEvent class is defined in the java.awt package.

**215. What is the immediate superclass of Menu?**

What is the immediate superclass of Menu? MenuItem

**216. 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.

**217. Which class is the immediate superclass of the MenuComponent class.**

Object

**218. 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.

**219. What is the difference between the Boolean & operator and the && operator?**

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

**220. Name three subclasses of the Component class.**

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

**221. What is the GregorianCalendar class?**

The GregorianCalendar provides support for traditional Western calendars.

**222. Which Container method is used to cause a container to be laid out and redisplayed?**

validate()

**223. What is the purpose of the Runtime class?**

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

**224. How many times may an object's finalize() method be invoked by the garbage collector?**

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

**225. What is the purpose of the finally clause of a try-catch-finally statement?**

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

**226. What is the argument type of a program's main() method?**

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

**227. Which Java operator is right associative?**

The = operator is right associative.

**228. 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.

**229. Can a double value be cast to a byte?**

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

**230. What is the difference between a break statement and a continue statement?**

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

**231. 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.

**232. What method is invoked to cause an object to begin executing as a separate thread?**

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

**233. Name two subclasses of the TextComponent class.**

TextField and TextArea

**234. What is the advantage of the event-delegation model over the earlier event-inheritance model?**

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

**235. Which containers may have a MenuBar?**

Frame

**236. How are commas used in the initialization and iteration parts of a for statement?**

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

**237. What is the purpose of the wait() , notify(), and notifyAll() methods?**

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

**238. What is an abstract method?**

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

**239. 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.

**240. What is the relationship between the Canvas class and the Graphics class?**

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

**241. What are the high-level thread states?**

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

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

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

**243. Can a Byte object be cast to a double value?**

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

**244. What is the difference between a static and a non-static inner class?**

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

**245. What is the difference between the String and StringBuffer classes?**

String objects are constants. StringBuffer objects are not.

**246. If a variable is declared as private, where may the variable be accessed?**

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

**247. What is an object's lock and which objects have locks?**

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

**248. What is the Dictionary class?**

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

**249. 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.

**250. 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.

**251. When can an object reference be cast to an interface reference?**

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

**252. What is the difference between a Window and a Frame?**

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

**253. Which class is extended by all other classes?**

The Object class is extended by all other classes.

**254. Can an object be garbage collected while it is still reachable?**

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

**255. Is the ternary operator written  $x : y ? z$  or  $x ? y : z$  ?**

It is written  $x ? y : z$ .

**256. What is the difference between the Font and FontMetrics classes?**

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

**257. How is rounding performed under integer division?**

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

**258. 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.

**259. 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.

**260. 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.

**261. 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.

**262. What is the SimpleTimeZone class?**

The SimpleTimeZone class provides support for a Gregorian calendar.

**263. What is the Map interface?**

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

**264. Does a class inherit the constructors of its superclass?**

A class does not inherit constructors from any of its super classes.

**265. For which statements does it make sense to use a label?**

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

**266. What is the purpose of the System class?**

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

**267. Which TextComponent method is used to set a TextComponent to the read-only state?**

setEditable()

**268. 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.

**269. Is &&= a valid Java operator?**

No, it is not.

**267. Name the eight primitive Java types.**

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

**268. Which class should you use to obtain design information about an object?**

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

**269. What is the relationship between clipping and repainting?**

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

**270. Is "abc" a primitive value?**

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

**271. What is the relationship between an event-listener interface and an event-adaptor class?**

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

**272. What restrictions are placed on the values of each case of a switch statement?**

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

**273. What modifiers may be used with an interface declaration?**

An interface may be declared as public or abstract.

**274. Is a class a subclass of itself?**

A class is a subclass of itself.

**275. What is the highest-level event class of the event-delegation model?**

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

**276. What event results from the clicking of a button?**

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

**277. 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.

**278. What is the difference between a while statement and a do statement?**

A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

**279. 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.

**280. What advantage do Java's layout managers provide over traditional windowing systems?**

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

**281. 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.

**282. What modifiers can be used with a local inner class?**

A local inner class may be final or abstract.

**283. What is the difference between static and non-static variables?**

A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

**284. 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.

**285. 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.

**286. Can an exception be rethrown?**

Yes, an exception can be rethrown.

**287. Which Math method is used to calculate the absolute value of a number?**

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

**288. 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.

**289. When does the compiler supply a default constructor for a class?**

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

**290. When is the finally clause of a try-catch-finally statement executed?**

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

**291. Which class is the immediate superclass of the Container class?**

Component

**292. If a method is declared as protected, where may the method be accessed?**

A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

**293. How can the Checkbox class be used to create a radio button?**

By associating Checkbox objects with a CheckboxGroup.

**294. Which non-Unicode letter characters may be used as the first character of an identifier?**

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

**295. What restrictions are placed on method overloading?**

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

**296. What happens when you invoke a thread's interrupt method while it is sleeping or waiting?**

When a task's interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.

**297. 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.

**298. What is the return type of a program's main() method?**

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

**299. Name four Container classes.**

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

**300. 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.

**301. What class of exceptions are generated by the Java run-time system?**

The Java runtime system generates RuntimeException and Error exceptions.

**302. What class allows you to read objects directly from a stream?**

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

**303. What is the difference between a field variable and a local variable?**

A field variable is a variable that is declared as a member of a class. A local variable is a variable that is declared local to a method.

**304. 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.

**305. 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.

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

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

**307. What is the difference between the JDK 1.02 event model and the event-delegation model introduced with JDK 1.1?**

The JDK 1.02 event model uses an event inheritance or bubbling approach. In this model, components are required to handle their own events. If they do not handle a particular event, the event is inherited by (or bubbled up to) the component's container. The container then either handles the event or it is bubbled up to its container and so on, until the highest-level container has been tried. In the event-delegation model, specific objects are designated as event handlers for GUI components. These objects implement event-listener interfaces. The event-delegation model is more efficient than the event-inheritance model because it eliminates the processing required to support the bubbling of unhandled events.

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

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

**309. Why are the methods of the Math class static?**

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

**310. What Checkbox method allows you to tell if a Checkbox is checked?**

getState()

**311. What state is a thread in when it is executing?**

An executing thread is in the running state.

**312. What are the legal operands of the instanceof operator?**

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

**313. How are the elements of a GridLayout organized?**

The elements of a GridLayout are of equal size and are laid out using the squares of a grid.

**314. What is 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.

**315. If an object is garbage collected, can it become reachable again?**

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

**316. 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.

**317. What classes of exceptions may be thrown by a throw statement?**

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

**318. What are E and PI?**

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

**319. Are true and false keywords?**

The values true and false are not keywords.

**320. What is a void return type?**

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

**321. What is the purpose of the enableEvents() method?**

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

**322. What is the difference between the File and RandomAccessFile classes?**

The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

**323. What happens when you add a double value to a String?**

The result is a String object.

**324. 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..

**325. Which package is always imported by default?**

The java.lang package is always imported by default.

**326. What interface must an object implement before it can be written to a stream as an object?**

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

**327. 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.

**328. 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.

**329. What is a compilation unit?**

A compilation unit is a Java source code file.

**330. What interface is extended by AWT event listeners?**

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

**331. What restrictions are placed on method overriding?**

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

**332. How can a dead thread be restarted?**

A dead thread cannot be restarted.

**333. 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.

**334. What is a layout manager?**

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

**335. Which arithmetic operations can result in the throwing of an `ArithmeticException`?**

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

**336. What are three ways in which a thread can enter the waiting state?**

A thread can enter the waiting state by invoking its `sleep()` method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's `wait()` method. It can also enter the waiting state by invoking its (deprecated) `suspend()` method.

**337. Can an abstract class be final?**

An abstract class may not be declared as final.

**338. 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.

**339. What happens if a try-catch-finally statement does not have a catch clause to handle an exception that is thrown within the body of the try statement?**

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

**340. 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.

**341. What is the difference between a `Scrollbar` and a `ScrollPane`?**

A `Scrollbar` is a Component, but not a Container. A `ScrollPane` is a Container. A `ScrollPane` handles its own events and performs its own scrolling.

**342. What is the difference between a public and a non-public class?**

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

**343. To what value is a variable of the boolean type automatically initialized?**

The default value of the boolean type is false.

**344. Can try statements be nested?**

Try statements may be nested.

**345. What is the difference between the prefix and postfix forms of the `++` operator?**

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

**346. What is the purpose of a statement block?**

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

**347. 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.

**348. What modifiers may be used with a top-level class?**

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

**349. 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.

**350. 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.

**351. 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.

**352. 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.

**353. 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.

**354. What methods are used to get and set the text label displayed by a Button object?**

getLabel() and setLabel()

**355. Which Component subclass is used for drawing and painting?**

Canvas

**356. What are synchronized methods and synchronized statements?**

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

**357. What are the two basic ways in which classes that can be run as threads may be defined?**

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

**358. What are the problems faced by Java programmers who don't use layout managers?**

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

**359. What is the difference between an if statement and a switch statement?**

The if statement is used to select among two alternatives. It uses a boolean expression to decide which alternative should be executed. The switch statement is used to select among multiple alternatives. It uses an int expression to determine which alternative should be executed.

---

## **JAVA QUESTION ONLY**

1. What are the differences between a List and a Set?
2. What are value objects and how are they used with EJBs?
3. How do you code a Singleton in Java?
4. What's a dynamic proxy class?
5. When should you call SwingUtilities.invokeLater()?
6. What are the differences between a Session Bean and an Entity Bean?
7. How do you read an XML file in Java?
8. Give an example of the Decorator pattern in the Java API.
9. Difference between an abstract class and interface
10. Inner classes
11. Difference between an inner class and a normal class
12. What is Entity Bean and Session Bean ?
13. What are the methods of Entity Bean?
14. How does Stateful Session bean store its state ?
15. Why does Stateless Session bean not store its state even though it has ejbActivate and ejbPassivate ?
16. What are the services provided by the container ?
17. Types of transaction ?
18. What is bean managed transaction ?
19. Why does EJB needs two interface( Home and Remote Interface) ?
20. What are transaction attributes ?
21. What is the difference between Container managed persistent bean and Bean managed persistent entity bean ?
22. What is J2EE ?
23. What is JTS ?
24. How many entity beans used and how many tables can u use in EJB project ?
25. What is scalable, portability in J2EE?
26. What is Connection pooling? Is it advantageous?
27. Method and class used for Connection pooling ?
28. How to deploy in J2EE(i.e Jar, War file) ?

29. How is entity bean created using Container managed entity bean ?
30. Software architecture of EJB ?
31. In Entity bean will the create method in EJB home and ejbCreate in Entity bean have the same parameters ?
32. What methods do u use in Servlet - Applet communication ?
33. What are the types of Servlet ?
34. Difference between HttpServlet and Generic Servlets ?
35. Difference between doGet and doPost ?
36. What are the methods in HttpServlet?
37. What are the types of SessionTracking?
38. What is Cookie ? Why is Cookie used ?
39. If my browser does not support Cookie, and my server sends a cookie instance What will happen ?
40. Why do u use Session Tracking in HttpServlet ?
41. Can u use javaScript in Servlets ?
42. What is the capacity the doGet can send to the server ?
43. What are the type of protocols used in HttpServlet ?
44. Difference between TCP/IP and IP protocol ?
45. Why do you use UniCastRemoteObject in RMI ?
46. How many interfaces are used in RMI?
47. Can Rmi registry be written in the code, without having to write it in the command prompt and if yes where?
48. Why is Socket used ?
49. What are the types of JDBC drivers ?
50. Explain the third driver(Native protocol driver) ?
51. Which among the four driver is pure Java driver ?
52. What are the Isolation level in JDBC transaction ?
53. How do you connect with the database ?
54. How do you connect without the Class.forName (" ") ?
55. What does Class.forName return ?
56. What are the types of statement ?
57. Why is preparedStatement, CallableStatement used for?
58. In real time project which driver did u use ?
59. Difference between AWT and Swing components ?
60. Is there any heavy weight component in Swings ?
61. Can the Swing application if you upload in net, be compatible with your browser?
62. What should you do get your browser compatible with swing components?
63. What are the methods in Applet ?
64. When is init(), start() called ?
65. When you navigate from one applet to another what are the methods called ?
66. What is the difference between Trusted and Untrusted Applet ?
67. What is Exception ?
68. What are the ways you can handle exception ?
69. When is try, catch block used ?
70. What is finally method in Exceptions ?
71. What are the types of access modifiers ?
72. What is protected and friendly ?
73. What are the other modifiers ?
74. Is synchronised modifier ?

75. What is meant by polymorphism ?
76. What is inheritance ?
77. What is method Overloading ? What is this in OOPS ?
78. What is method Overriding ? What is it in OOPS ?
79. Does java support multi dimensional arrays ?
80. Is multiple inheritance used in Java ?
81. How do you send a message to the browser in JavaScript ?
82. Does javascript support multidimensional arrays ?
83. Why is XML used mainly?
84. Do use coding for database connectivity in XML?
85. What is DTD ?
86. Is there any tool in java that can create reports ?
87. Explain protected vs private vs public data members and methods?
88. Explain overloading and overriding?
89. What is polymorphism?
90. What is inheritance?
91. What is encapsulation?
92. Give an example of where you might use a static method?
93. Explain up casting and down casting?
94. What is an abstract class?
95. Explain multiple inheritance and interfaces?
96. If you have a tree of derived classes, what order are the constructors run
97. How do you deal with errors?
98. What is a ClassLoader?
99. How are ClassLoaders used, when and why?
100. What is a fully qualified class name?
101. What is the classpath, and how does it relate to classloaders?
102. When does the JVM invoke a classloader?
103. When invoked, how does a classloader handle the loading of a class?
104. Can you pass classes between classloader, why/whynot?
105. What is a ClassDefNotFoundError?
106. What is a ClassCastException?
107. What is a CircularityError?
108. How are classes stored in the JVM?
109. Can CORBA be integrated with XML?
110. can I make my site completely safe by running the server in a "chroot" environment?
111. Whats the "chroot" environment?
112. How can I detect if my site's been broken into?
113. what types of access restrictions are available?
114. Isn't there a CGI Script to allow users to change their passwords online?
115. What are: SSL, SHTTP, Shen? and Cybercash, SET and Open Market?
116. Are there any "freeware" secure services?
117. Can I use Personal Certificates to control server access?
118. What are Perl taint checks? How do I turn them On?
119. How do I untaint a variable?
120. How can you push data from an Applet to Servlet ?
121. How you can know about drivers and database information ?

122. If you are truncated using JDBC, How can you know ..that how much data is truncated ?
123. And What situation , each of the 4 drivers used ?
124. In RMI, server object first loaded into the memory and then the stub reference is sent to the client ? or whether a stub reference is directly sent to the client ? Suppose server object is not loaded into the memory, and the client request for it , what will happen?
125. Can you load the server object dynamically? If so, what are the major 3 steps involved in it ?
126. What is difference RMI registry and OSAgent ?
127. What is the corresponding Layout for Card in Swing ?
128. What is light weight component ?
129. Can you run the product development on all operating systems ?
130. What is difference in between Java Class and Bean ?
131. Can we send object using Sockets ?
132. How to communicate 2 threads each other ?
133. What are the files generated after using IDL to Java Compilet ?
134. Can I modify an object in CORBA ?
135. What is the functionality stubs and skeletons ?
136. What is the mapping mechanism used by Java to identify IDL language ?
137. Suppose If we have variable ' I ' in run method, If I can create one or more thread each thread will occupy a separate copy or same variable will be shared ?
138. In servlets, we are having a web page that is invoking servlets username and password ? which is cheks in the database ? Suppose the second page also If we want to verify the same information whethe it will connect to the database or it will be used previous information?
139. What are session variable in Servlets ?
140. How will you pass parameters in RMI ? Why u serialize?
141. What is the exact difference in between Unicast and Multicast object ? Where we will use ?
142. What is the main functionality of the Remote Reference Layer ?
143. How do you download stubs from a Remote place ?
144. I want to store more than 10 objects in a remote server ? Which methodology will follow ? What is meant by Session ? Tell me something about HttpSession Class ?
145. Have you used threads in Servlet ?
146. How do you sing an Applet ?
147. In a Container there are 5 components. I want to display the all the components names, how will you do that one ?
148. Why there are some null interface in java ? What does it mean ? Give me some null interfaces in JAVA ?
149. Tell me the latest versions in JAVA related areas ?
150. What is meant by class loader ? How many types are there? When will we use them ?
151. How do you load an Image in a Servlet ?
152. What is meant by flickering ?
153. What is meant by distributed Application ? Why we are using that in our applications ?

- 154. What is the functionality of the stub ?
- 155. What is the latest version of JDBC ? What are the new features are added in that ?
- 156. What is the role of the webserver ?
- 157. What is meant by cookies ? Explain ?
- 158. How OOPS concept is achieved in Java
- 159. How does Java 2.0 differ from Java 1.0
- 160. What are command line arguments
- 161. Wrapper class. Is String a Wrapper Class
- 162. What are the restriction for static method
- 163. Purpose of the file class