Core Java Interview Questions

1. Why threads block or enters to waiting state on I/O?
Threads enters to waiting state or block on I/O because other threads can execute while the I/O operations are performed.

2. What is List interface?
List is an ordered collection of objects.

3. What is the difference between yield() and sleep()?
When a object invokes yield() it returns to ready state. But when an object invokes sleep() method enters to not ready state.

8. What are Wrapper Classes ?
They are wrappers to primitive data types. They allow us to access primitives as objects.

9. Can we call finalize() method ?
Yes. Nobody will stop us to call any method , if it is accessible in our class. But a garbage collector cannot call an object's finalize method if that object is reachable.

10. What is the difference between time slicing and preemptive scheduling ?
In preemptive scheduling, highest priority task continues execution till it enters a not running state or a higher priority task comes into existence. In ime slicing, the task continues its execution for a predefined period of time and reenters the pool of ready tasks.

11. What is the initial state of a thread when it is created and started?
The thread is in ready state.

12. Can we declare an anonymous class as both extending a class and implementing an interface?
No. An anonymous class can extend a class or implement an interface, but it cannot be declared to do both

13. What are the differences between boolean && operator and & operator?
When an expression containing the & operator is evaluated, both operands are evaluated. And the & operator is applied to the operand. When an expression containing && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then only the second operand is evaluated otherwise the second part will not get executed. && is also called short cut and.

14.What is the use of the finally block?
Finally is the block of code that executes always. The code in finally block will execute even if an exception is occurred. finally will not execute when the user calls System.exit().

15. What is an abstract method ?
An abstract method is a method that don't have a body. It is declared with modifier abstract.
An abstract method is a method whose implementation is deferred to a subclass.

16. What is the difference between System.err and System.out?
We can redirect System.out to another file but we cannot redirect System.err stream

17. What are the differences between an abstract class and an interface?
An abstract class can have concrete method, which is not allowed in an interface. Abstract class can have private or protected methods and variables and only public methods and variables are allowed in interface. We can implement more than one interface , but we can extend only one abstract class. Interfaces provides loose coupling where as abstract class provides tight coupling.

18. What is the difference between synchronized block and synchronized method?
Synchronized blocks place locks for the specified block where as synchronized methods place locks for the entire method.

19. How can you force garbage collection in java?
You cannot force Garbage Collection, but you can request for it by calling the method System.gc(). But it doesn't mean that Garbage Collection will start immediately. The garbage collection is a low priority thread of JVM.

20. How can you call a constructor from another constructor ?
By using this() reference.

21. How can you call the constructor of super class ?
By using super() syntax.

22. What's the difference between normal methods and constructors?
Constructors must have the same name of the class and can not have a return type. They are called only once, while regular methods can be called whenever required. We cannot explicitly call a constructor.

23. What is the use of packages in java ?
Packages are a way to organize files in java when a project consists of more than one module. It helps in resolving name conflicts when different modules have classes with the same names.

24. What must be the order of catch blocks when catching more than one exception?

The sub classes must come first. Otherwise it will give a compile time error.

25. How can we call a method or variable of the super class from child class ?

We can use super.method() or super.variable syntax for this purpose.

26. If you are overriding equals() method of a class, what other methods you might need to override ?

hashCode

27. How can you create your own exception?

Our class must extend either Exception or its sub class

28. What is serialization?

Serialization is the process of saving the state of an object.

29. What is de-serialization?

De-serialization is the process of restoring the state of an object.

30. What is externalizable?

It is an interface that extends Serializable. It is having two different methods writeExternal() and readExternal. This interface allows us to customize the output.

31. 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 also possible for programs to create objects that are not subject to garbage collection. And there is no guarantee that Garbage Collection thread will be executed.

32. What is a native method?

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

33. What are different type of exceptions in Java?

There are two types of exceptions in java. Checked exceptions and Unchecked exceptions. Any exception that is is derived from Throwable and Exception is called checked exception except RuntimeException and its sub classes. The compiler will check whether the exception is caught or not at compile time. We need to catch the checked exception or declare in the throws clause. Any exception that is derived from Error and RuntimeException is called unchecked exception. We don't need to explicitly catch a unchecked exception.

34. Can we catch an error in our java program ?

Yes. We can . We can catch anything that is derived from Throwable. Since Error is a sub class of Throwable we can catch an error also.

35. What is thread priority?

Thread Priority is an integer value that identifies the relative order in which it should be executed with respect to others. The thread priority values ranging from 1- 10 and the default value is 5. But if a thread have higher priority doesn't means that it will execute first. The thread scheduling depends on the OS.

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

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

Break statement results in the immediate 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.

38. What must a class do to implement an interface?

It must identify the interface in its implements clause. Also it must provide definition for all the methods in the interface otherwise it must be declared abstract.

39. What is an abstract class?

An abstract class is an incomplete class. It is declared with the modifier abstract. We cannot create objects of the abstract class. It is used to specify a common behavioral protocol for all its child classes.

40. What is the difference between notify and notifyAll method ?

notify wakes up a single thread that is waiting for object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation. notifyAll Wakes up all threads that are waiting on this object's monitor. A thread waits on an object's monitor by calling one of the wait methods.

41. What does wait method do ?

It causes current thread to wait until either another thread invokes notify or notifyAll method of the current object, or a specified amount of time has elapsed.

42. What are the different states of a thread?

The different thread states are ready, running, waiting and dead.

43. What is the difference between static and non static inner class?

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

44. What is the difference between String and StringBuffer class ?

Strings are immutable (constant), their values cannot be changed after they are created. StringBuffer supports mutable objects.

45. Which is the base class for all classes?

java.lang.Object.

46. What is the difference between readers and streams?

Readers are character oriented where streams are byte oriented. The readers are having full support for Unicode data.

47. What is constructor chaining?

When a constructor of a class is executed it will automatically call the default constructor of the super class (if no explicit call to any of the super class constructor) till the root of the hierarchy.

48. What are the different primitive data types in java?

There are 8 primitive types in java. boolean , char, byte, short, int long, float, double.

49. What is static?

static means one per class. static variables are created when the class loads. They are associated with the class. In order to access a static we don't need objects. We can directly access static methods and variable by calling classname.variablename.

50. Why we cannot override static methods?

Static means they are associated with a class. In static methods, the binding mechanism is static binding. So it must be available at the compile time.

51. 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. There will be only one value for static variable for all instances of that class. Non-static variables take on unique values with each object instance.

52. When does a compiler supplies a default constructor for a class?

If there is no other constructor exist in a class, the compiler will supply a default constructor.

53. What are the restrictions placed on overriding a method ?

The overridden method have the exact signature of the super class method, including the return type. The access specified cannot be less restrictive than the super class method. We cannot throw any new exceptions in overridden method.

54. What are the restrictions placed on overloading a method?

Overloading methods must differ in their parameter list, or number of parameters.

55. What is casting?

Casting means converting one type to another. There are mainly two types of casting. Casting between primitive types and casting between object references.

Casting between primitive numeric types is used to convert larger data types to smaller data types. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

56. What is the difference between == and equals?

The equals method can be considered to perform a deep comparison of the value of an object, whereas the == operator performs a shallow comparison. If we are not overriding the equals method both will give the same result. == will is used to compare the object references. It is used to check whether two objects are points to the same reference.

57. What is a void return type ?

void indicates that the method will not return anything.

58. What will happen if an exception is not caught ?

An uncaught exception results in the uncaughtException() method of the thread's ThreadGroup, which results in the termination of the program.

59. What are the different ways in which a thread can enter into waiting state?

There are three ways for a thread to enter into 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.

60. What is a ResourceBundle class?

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

61. What is numeric promotion?

Numeric promotion is the conversion of a smaller numeric type to a larger numeric type. In numerical promotion, byte, char, and short values are converted to int values. The int, long and float values are converted to the desired types if required.

62. What is the difference between the prefix and postfix forms of the ++ operator?

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

63. What are synchronized methods and synchronized statements?

Synchronized methods are methods that are declared with the keyword synchronized. A thread executes a synchronized method only after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. It is a block of code declared with synchronized keyword.

A synchronized statement can be executed only after a thread has acquired the lock for the object or class referenced in the synchronized statement.

64. How can we create a thread?

thread can be created by extending Thread class or by implementing Runnable interface. Then we need to override the method public void run().

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

If statement is used to select from two alternatives. It uses a boolean expression to decide which alternative should be executed. The expression in if must be a boolean value. The switch statement is used to select from multiple alternatives. The case values must be promoted to an to int value.

66. What is hashCode?

The hashcode of a Java Object is simply a number (32-bit signed int) that allows an object to be managed by a hash-based data structure. A hashcode should
be, equal for equal object (this is mandatory!) , fast to compute based on all or most of the internal state of an object, use all or most of the space of
32-bit integers in a fairly uniform way , and likely to be different even for objects that are very similar. If you are overriding hashCode you need to override equals method also.

67. 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.

68. What is the difference between RandomAccessFile and File?

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

69. What is final ?

final is a keyword in java. If final keyword is applied to a variable, then the variable will become a constant. If it applied to method, sub classes cannot override the method. If final keyword is applied to a class we cannot extend from that class.

70. What is the difference among JVM Spec, JVM Implementation, JVM Runtime?

The JVM spec is the blueprint for the JVM generated and owned by Sun. The JVM implementation is the actual implementation of the spec by a vendor and the JVM runtime is the actual running instance of a JVM implementation

71. How is the difference between thread and process?

process runs in its own address space. No two processes share their address space. Threads will run in the same address space of the process that owns them.

72. What is the difference between Vector and ArrayList ?

Vector is synchronized, ArrayList is not. Vector is having a constructor to specify the incremental capacity. But ArrayList don't have. By default Vector grows by 100% but ArrayList grows by 50% only.

73. What is the difference between Hashtable and HashMap ?

Hashtable is synchronized . but HashMap is not synchronized. Hashtable does not allow null values , but HashMap allows null values.

74. What are the access modifiers available in Java.

Access modifier specify where a method or attribute can be used. Public is accessible from anywhere. Protected is accessible from the same class and its subclasses. Package/Default are accessible from the same package. Private is only accessible from within the class.

75. Why java is said to be pass-by-value ?

When assigning an object to a variable, we are actually assigning the memory address of that object to the variable. So the value passed is actually the memory location of the object. This results in object aliasing, meaning you can have many variables referring to the same object on the heap.

76. What do you mean by immutable ? How to create an immutable object?

Immutability means an object cannot be modified after it has been initialized. There will not be any setter methods in an immutable class. And normally these classes will be final.

77. What is class loader in java ?

class loader is a class that is responsible for loading the class. All JVM contains one class loader called primordial class loader.

78. What is a weak reference?

weak reference is the one that does nor prevent the referenced object from being garbage collected. The weak reference will not keep the object that it refers to alive. A weak reference is not counted as a reference in garbage collection. This will make the memory use more effective.

79. What is object cloning?

It is the process of duplicating an object so that two identical objects will exist in the memory at the same time.

80. What is object pooling?

Creating a large number of identical short lived objects is called object pooling. This helps to minimize the need of garbage collection and makes the memory use more effective.

81. What is garbage collection?

Garbage collection is the process of releasing memory used by unreferenced objects. It relieves the programmer from the process of manually releasing the memory used by objects .

82. What is the disadvantage of garbage collection?

It adds an overhead that can affect performance. Additionally there is no guarantee that the object will be garbage collected.

83. What is a Dictionary?

Dictionary is a parent class for any class that maps keys to values. In a dictionary every key is associated with at most one value.

84. What is JAR file ?

JAR stands for Java Archive. This is a file format that enables you to bundle multiple files into a single archive file. A jar file will contains a manifest.mf file inside META-INF folder that describes the version and other features of jar file.

85. Why Java is not fully objective oriented ?

Due to the use of primitives in java, which are not objects.

86. What is a marker interface ?

An interface that contains no methods. Eg: Serializable, Cloneable, SingleThreadModel etc. It is used to just mark java classes that support certain capability.

87. What are tag interfaces?

Tag interface is an alternate name for marker interface.

88. What are the restrictions placed on static method ?

We cannot override static methods. We cannot access any object variables inside static method. Also the this reference also not available in static methods.

89. What is JVM?

JVM stands for Java Virtual Machine. It is the run time for java programs. All are java programs are running inside this JVM only. It converts java byte code to OS specific commands. In addition to governing the execution of an application's byte codes, the virtual machine handles related tasks such as managing the system's memory, providing security against malicious code, and managing multiple threads of program execution.

90. What is JIT?

JIT stands for Just In Time compiler. It compiles java byte code to native code.

91. What is java byte code?

Byte code is an sort of intermediate code. The byte code is processed by virtual machine.

92. What is method overloading?

Method overloading is the process of creating a new method with the same name and different signature.

93. What is method overriding?

Method overriding is the process of giving a new definition for an existing method in its child class.

94. What is finalize() ?

Finalize is a protected method in java. When the garbage collector is executes , it will first call finalize( ), and on the next garbage-collection it reclaim the objects memory. So finalize( ), gives you the chance to perform some cleanup operation at the time of garbage collection.

95. What is multi-threading?

Multi-threading is the scenario where more than one threads are running.

96. What is deadlock?

Deadlock is a situation when two threads are waiting on each other to release a resource. Each thread waiting for a resource which is held by the other waiting thread.

97. What is the difference between Iterator and Enumeration?

Iterator differ from enumeration in two ways Iterator allow the caller to remove elements from the underlying collection during the iteration with well-defined semantics. And , method names have been improved.

98. What is the Locale class?

Locale object represents a specific geographical, political, or cultural region

99. What is internationalization?

Internationalization is the process of designing an application so that it can be adapted to various languages and regions without changes.

100. What is anonymous class ?

An anonymous class is a type of inner class that don't have any name.

101. What is the difference between URL and URLConnection?

URL represents the location of a resource, and a URLConnection represents a link for accessing or communicating with the resource at the location.

102. What are the two important TCP Socket classes?

ServerSocket and Socket. ServerSocket is useful for two-way socket communication. Socket class help us to read and write through the sockets.

getInputStream() and getOutputStream() are the two methods available in Socket class.

103. Strings are immutable. But String s="Hello"; String s1=s+"World" returns HelloWorld how?

Here actually a new object is created with the value of HelloWorld

106. What is java collections?

Java collections is a set of classes, that allows operations on a collection of classes.

107. Can we compile a java program without main?

Yes, we can. In order to compile a java program, we don't require any main method. But to execute a java program we must have a main in it (unless it is an applet or servlet). Because main is the starting point of a java program.

108. What is a java compilation unit.

A compilation unit is a java source file.

109. What are the restrictions when overriding a method ?

Overridden methods must have the same name, argument list, and return type (i.e., they must have the exact signature of the method we are going to override, including return type.) The overriding method cannot be less visible than the method it overrides(i.e., a public method cannot be override to private). The overriding method may not throw any exceptions that may not be thrown by the overridden method

110. What is static initializer block? What is its use?

A static initializer block is a block of code that declares with the static keyword. It normally contains the block of code that must execute at the time of class loading. The static initializer block will execute only once at the time of loading the class only.

111. How does a try statement determine which catch clause should be used to handle an exception?

When an exception is thrown , the catch block of the try statement are examined in the order in which they appear. The first catch block that is capable of handling the exception is executed. The remaining catch blocks are ignored

112. How parameters are passed to methods in java program ?

All java method parameters in java are passed by value only. Obviously primitives are passed by value. In case of objects a copy of the reference is passed and so all the changes made in the method will persist.

113. If a class doesn't have any constructors, what will happen?

If a class doesn't have a constructor, the JVM will provide a default constructor for the class.

114. What will happen if a thread cannot acquire a lock on an object?

It enters to the waiting state until lock becomes available.

115. How does multithreading occurring on a computer with a single CPU?

The task scheduler of OS allocates an execution time for multiple tasks. By switching between different executing tasks, it creates the impression that tasks execute sequentially. But actually there is only one task is executed at a time.

116. What will happen if you are invoking a thread's interrupt method while the thread is waiting or sleeping?

When the task enters to the running state, it will throw an InterruptedException.

117. What are the different ways in which a thread can enter into waiting state?

There are three ways for a thread to enter into 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.

118. What are the the different ways for creating a thread?

A thread can be created by subclassing Thread, or by implementing the Runnable interface.

119. What is the difference between creating a thread by extending Thread class and by implementing Runnable interface? Which one should prefer?

When creating a thread by extending the Thread class, it is not mandatory to override the run method (If we are not overriding the run method , it is useless), because Thread class have already given a default implementation for run method. But if we are implementing Runnable , it is mandatory to override the run method. The preferred way to create a thread is by implementing Runnable interface, because it give loose coupling.

120. What is coupling?

A Coupling is the dependency between different components of a system

136. Why Java is not 100% pure object oriented language?

Because java uses primitives.

137. Why ArrayList is faster than Vector?

Because Vector is synchronized. Synchronization reduces the performance.

138. What is the security mechnaism used in java?

Java uses sand box security model.

139. What is sandbox?

A sandbox is a security mechanism for safely running programs. The sandbox typically provides a tightly-controlled set of resources for guest programs to run in, such as scratch space on disk and memory.

140. What is phantom memory?

Phantom memory is the memory that does not exist in reality.

141. What is reflection?

Reflection is the process of finding out the different features of a class dynamically.

142. What are the differences between JIT and HotSpot?

The Hotspot VM is a collection of techniques, the most important of which is called adaptive optimization. The original JVMs interpreted byte codes one at a time. Second-generation JVMs added a JIT compiler, which compiles each method to native code upon first execution, then executes the native code. Thereafter, whenever the method is called, the native code is executed. The adaptive optimization technique used by Hotspot is a hybrid approach, one that combines byte code interpretation and run-time compilation to native code. Hotspot, unlike a regular JIT compiling VM, doesn't do "premature optimization"

143. What are the advantages and disadvantages of reference counting in garbage collection?

An advantage of this scheme is that it can run in small chunks of time closely linked with the execution of the program. These characteristic makes it particularly suitable for real-time environments where the program can't be interrupted for very long time. A disadvantage of reference counting is that it does not detect cycles. A cycle is two or more objects that refer to one another. Another disadvantage is the overhead of incrementing and decrementing the reference count each time. Because of these disadvantages, reference counting currently is out of favor.

144. How would you implement a thread pool?

The ThreadPool class is a generic implementation of a thread pool, which takes the following input Size of the pool to be constructed and name of the class which implements Runnable (which has a visible default constructor) and constructs a thread pool with active threads that are waiting for activation. once the threads have finished processing they come back and wait once again in the pool.

145. What is the difference between throw and throws clause?

A throw is used to throw an exception manually, where as throws is used in the case of checked exceptions, to tell the compiler that we haven't handled the exception, so that the exception will be handled by the calling function.

147. What is a classloader?

A class loader is an object that is responsible for loading classes.

148. What is the difference between Comparable and Comparator ?

A The Comparable is for natural ordering and Comparator is for custom ordering. But we can override the compareTo method of comparable interface to give a custom ordering.

149. What is the difference between List, Set and Map?

A A Set is a collection that has no duplicate elements. A List is a collection that has an order associated with its elements. A map is a way of storing key/value pairs. The way of storing a Map is similar to two-column table.

150. What is the difference between Exception and Error?

Error is unrecoverable.

151. What is meant by Open Source?

In general, open source refers to any program whose source code is made available for use or modification as users or other developers see fit. Open source software is usually developed as a public collaboration and made freely available.

152. How do you send data from an applet to Servlet ? What are the steps involved in it?

You can use the java.net.URLConnection and java.net.URL classes to open a standard HTTP connection to the web server. The server then passes this information to the servlet in the normal way. Basically, the applet pretends to be a web browser, and the servlet doesn't know the difference. As far as the servlet is concerned, the applet is just another HTTP client.

The Package Young man mails himself in a box to his girlfriend... naked. Drug Rehab Helpful links on drug addictions, treatment and rehab. Forex Helpful links on Forex currency trading. Tired Of MySpace? Meet cool people at TagWorld!

153. What is polymorphism?

It is the ability of an object to behave differently on different situations for the same message.

154. What is a class, member and local variable?

Variables declared within a method are local variables. Variables declared within the class are member variables. Variables declared within the class with static modifier are class variables

155. How do I convert a numeric IP address like 66.29.36.130 into a hostname like www.javacertificate.net

String hostname = InetAddress.getByName("66.29.36.130").getHostName();

156. What is the difference between a constructor and a method?

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator. We cannot invoke a constructor directly. A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

157. What are the different inner classes types?

There are mainly four types available. They are Member classes, Nested top-level classes, Local classes, Anonymous classes

158. What is Nested top-level classes?

A class declared within a class with static modifier is called nested top level class. Any class outside the declaring class can access the nested top level class with the declaring class dot nested top level class. Top-level inner classes have access to static variables only .

159. What is Member classes?

A class declared inside a class without static modifier is called member class. Member classes are just like any other member methods or member variables.

160. What is Local inner classes?

Local inner classes are class declared inside a block of code. They are visible only within the block of their declaration.

161. Can a top level class be private or protected?

No. A top level class can not be private or protected. It can have either "public" or no modifier.

162. How will you invoke any external process in Java?

A By using Runtime.getRuntime().exec(….)

163. What is a package?

A To group set of classes into a single unit is known as packaging. Packages provides wide namespace visibility.

164. What is the use of assert keyword

A Assert keyword validates certain expressions. It replaces the if block effectively and throws an AssertionError on failure. The assert keyword should be used only for critical arguments (means without that the method does nothing).

165. What is composition?

A Holding the reference of the other class within some other class is known as composition.

166. What is aggregation?

It is a special type of composition. If you expose all the methods of a composite class and route the method call to the composite method through its reference, then it is called aggregation

167. What are the methods in Object?

clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString

168. 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. 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.

169. What factors are used to decide using synchronized or volatile?

You can make a variable synchronized or volatile under the following cases: 1) if you are not updating many variables often in a multithread environment,
consider using volatile. 2) If you are updating many variables, consider using synchronized, because using volatile might be slower.

170. 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.

171. What is the difference between static synchronized and synchronized methods?

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.

172. What is the purpose of the Runtime class?

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

173. What is the purpose of the System class?

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

174. 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.

175. Considering notepad/IE or any other thing as process, What will happen if you start notepad or IE 3 times? Where 3 processes are started or 3 threads are started ?

3 processes will start.

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

A At compile time, each case values of switch statement must evaluate to a an int value.

177. If aaaa is an array then why aaaa.length why not aaaa.length()?

Because length is a property not a method.

178. What is dynamic typing?

A Dynamic typing means type errors are detected at run time by checking the actual data types of the values against the required data types

179. What is static typig?

A Static typing means type errors are detected at compile time by checking the inferred data type is a subtype of the required type

180. What is HashMap and Map?

A Map is Interface and HashMap is class that implements that.

181. What is an Object and how do you allocate memory to it?

Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

182. What is UNICODE?

Unicode is used for internal representation of characters and strings and it uses 16 bits to represent each other.

183. What is adapter class?

An adapter class provides a default implementation of all methods in an event listener interface. Adapter classes are useful when you want to process only some of the events that are handled by a particular event listener interface. You can define a new class by extending one of the adapter classes and implementing only those events relevant to us.

VoIP Service Info and helpful links on VoIP service and systems. The Package Young man mails himself in a box to his girlfriend... naked. Web Hosting Helpful links on on web hosting, including reviews and more. Online Basketball Manager Take controll of your free team and take it to the top!

184. What is a stream and what are the types of Streams and classes of the Streams?

A stream is a flow of data from one direction to another. . There are two types of Streams Byte Streams: Provide a convenient means for handling input and output of bytes. And Character Streams: Provide a convenient means for handling input & output of characters.

185. What is the difference between TCP/IP and UDP?

TCP/IP is a two-way communication between the client and the server and it is a reliable and there is a confirmation regarding reaching the message to the destination. UDP is a one-way communication only between the client and the server and it is not a reliable and there is no confirmation regarding reaching the message to the destination.

186. What is Inter-Thread communication?.

Exchange of information between two threads.

187. What is a policy?.

It's an abstract class for representing the system security policy for a Java application environment (specifying which permissions are available for code from various sources). Java security properties file resides in <JAVA-HOME>/lib/security/java.security directory.

188. What is a thread group?
A thread group is a data structure that controls the state of collection of thread as a whole managed by the particular runtime environment.
189. Why is UTFDataFormatException thrown by DataOutputStream.writeUTF() when serializing a String?
DataOutputStream.writeUTF() does not support writing out strings larger than 64K. The first two bytes of a UTF string in the stream are the length of the string. If a java.lang.String is larger than 64K, it needs to be stored in the stream by an alternative method rather than depending on the default method of storing a String in the stream, writeUTF.
190. Why is OutOfMemoryError thrown after writing a large number of objects into an ObjectOutputStream?
The ObjectOutputStream maintains a table mapping objects written into the stream to a handle. The first time an object is written to a stream, its contents are written into the stream; subsequent writes of the object result in a handle to the object being written into the stream. This table maintains references to objects that might otherwise be unreachable by an application, thus, resulting in an unexpected situation of running out of memory. A call to the ObjectOutputStream.reset() method resets the object/handle table to its initial state, allowing all previously written objects to be eligible for garbage collection.
191. How can I get the serialVersionUID of a class?
By running the serialver tool with the name of the class as the command line argumet, as shown in the example that follows: serialver java.lang.String
192. What is serialVersionUID ?
A The serialVersionUID is a universal version identifier for a Serializable class. Deserialization uses this identifier number to ensure that a loaded class corresponds to a serialized object.
193. What is abstraction?
An abstraction is an idea, concept, or word which defines the phenomena which make up the concrete events or things which the abstraction refers to, the referents.
194. What is encapsulation?
A Encapsulation describes the ability of an object to hide its data and methods from the rest of the world
195. What is inheritance?
Inheritance is the ability to create new classes based on existing classes. It is useful to reuse existing code.
195. How could Java classes direct program messages to the system console, but error messages, say to a file?
A. The class System has a variable out that represents the standard output, and the variable err that represents the standard error device. By default, they both point at the system console. This how the standard output could be re-directed:
Stream st = new Stream(new FileOutputStream("output.txt")); System.setErr(st); System.setOut(st);
196. What's the difference between an interface and an abstract class?
A. An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.
197. Why would you use a synchronized block vs. synchronized method?
A. Synchronized blocks place locks for shorter periods than synchronized methods.
4. Explain the usage of the keyword transient?
A. This keyword indicates that the value of this member variable does not have to be serialized with the object. When the class will be de-serialized, this variable will be initialized with a default value of its data type (i.e. zero for integers).
198. How can you force garbage collection?
A. You can't force GC, but could request it by calling System.gc(). JVM does not guarantee that GC will be started immediately.

199. How do you know if an explicit object casting is needed?
A. If you assign a superclass object to a variable of a subclass's data type, you need to do explicit casting. For example:

Object a; Customer b; b = (Customer) a;
When you assign a subclass to a variable having a supeclass type, the casting is performed automatically.

200. What's the difference between the methods sleep() and wait()
A. The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.
2001. Can you write a Java class that could be used both as an applet as well as an application?
Yes. Add a main() method to the applet.
2002. What's the difference between constructors and other methods?
A. Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.
203. Can you call one constructor from another if a class has multiple constructors
A. Yes. Use this() syntax.

204. Explain the usage of Java packages.
A. This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.
205. If a class is located in a package, what do you need to change in the OS environment to be able to use it?
A. You need to add a directory or a jar file that contains the package directories to the CLASSPATH environment variable. Let's say a class Employee belongs to a package com.xyz.hr; and is located in the file c:\dev\com\xyz\hr\Employee.java. In this case, you'd need to add c:\dev to the variable CLASSPATH. If this class contains the method main(), you could test it from a command prompt window as follows:

c:\>java com.xyz.hr.Employee

206. What's the difference between J2SDK 1.5 and J2SDK 5.0?
A.There's no difference, Sun Microsystems just re-branded this version.
206. What would you use to compare two String variables - the operator == or the method equals()?
A. I'd use the method equals() to compare the values of the Strings and the == to check if two variables point at the same instance of a String object.
207. Does it matter in what order catch statements for FileNotFoundException and IOExceptipon are written?
A. Yes, it does. The FileNoFoundException is inherited from the IOException. Exception's subclasses have to be caught first.
207. Can an inner class declared inside of a method access local variables of this method?
It's possible if these variables are final.
208. What can go wrong if you replace && with & in the following code:
String a=null; if (a!=null && a.length()>10) {...}

A single ampersand here would lead to a NullPointerException.
210. When should the method invokeLater()be used?
A. This method is used to ensure that Swing components are updated through the event-dispatching thread.

211. How can a subclass call a method or a constructor defined in a superclass?
Use the following syntax: super.myMethod(); To call a constructor of the superclass, just write super(); in the first line of the subclass's constructor.
212. What's the difference between a queue and a stack?
A. Stacks works by last-in-first-out rule (LIFO), while queues use the FIFO rule
213. You can create an abstract class that contains only abstract methods. On the other hand, you can create an interface that declares the same methods. So can you use abstract classes instead of interfaces?
A. Sometimes. But your class may be a descendent of another class and in this case the interface is your only option.
214. What comes to mind when you hear about a young generation in Java?
A. Garbage collection.
215. What comes to mind when someone mentions a shallow copy in Java?
A. Object cloning.
216. If you're overriding the method equals() of an object, which other method you might also consider?
A. hashCode()
217. You are planning to do an indexed search in a list of objects. Which of the two Java collections should you use:
ArrayList or LinkedList?
A. ArrayList
218. How would you make a copy of an entire Java object with its state?
Have this class implement Cloneable interface and call its method clone().
219. How can you minimize the need of garbage collection and make the memory use more effective?
A. Use object pooling and weak object references.
220. There are two classes: A and B. The class B need to inform a class A when some important event has happened. What Java technique would you use to implement it?
A. If these classes are threads I'd consider notify() or notifyAll(). For regular classes you can use the Observer interface.
221. What access level do you need to specify in the class declaration to ensure that only classes from the same directory can access it?
A. You do not need to specify any access level, and Java will use a default package access level.

222. What is the diffrence between an Abstract class and Interface?

- Abstract classes may have some executable methods and methods left unimplemented. Interfaces contain no implementation code.
- A class can implement any number of interfaces, but subclass at most one abstract class.
- An abstract class can have nonabstract methods. All methods of an interface are abstract.

- An abstract class can have instance variables. An interface cannot.
- An abstract class can define constructor. An interface cannot.
- An abstract class can have any visibility: public, protected, private or none (package). An interface's visibility must be public or none (package).
- An abstract class inherits from Object and includes methods such as clone() and equals().

223. What is a user defined exception?

User-defined exceptions may be implemented by

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

The developer can define a new exception by deriving it from the Exception class as follows:

```
public class  MyException  extends  Exception  {

/* class definition of constructors (but NOT the exception handling code) goes here */

            public  MyException()  {

                        super();

            }

            public  MyException( String  errorMessage )  {
                        super(  errorMessage  );
            }
}
```

The throw statement is used to signal the occurance of the exception within a try block. Often, exceptions are instantiated in the same statement in which they are thrown using the syntax.

```
throw new MyException("I threw my own exception.")
```

To handle the exception within the method where it is thrown, a catch statement that handles MyException, must follow the try block. If the developer does not want to handle the exception in the method itself, the method must pass the exception using the syntax:

```
public myMethodName() throws MyException
```

2. What are statements in JAVA ?

Statements are equivalent to sentences in natural languages. A statement forms a complete unit of execution. The following types of expressions can be made into a statement by terminating the expression with a semicolon

- Assignment expressions
- Any use of ++ or --
- Method calls
- Object creation expressions

These kinds of statements are called expression statements.
In addition to these kinds of expression statements, there are two other kinds of statements. A declaration statement declares a variable. A control flow statement regulates the order in which statements get executed. The for loop and the if statement are both examples of control flow statements.

4. What is JNI ?

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

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

Disadvantages:

- You can®□□t say write once run anywhere.
- Difficult to debug runtime error in native code.
- Potential security risk.
- You can®□□t call it from Applet.

5. What is serialization ?

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

stream and ObjectInputStream to read it from the stream.

6. Why there are some null interface in java ? What does it mean ? Give me some null interfaces in JAVA?
Null interfaces act as markers..they just tell the compiler that the objects of this class need to be treated differently..some marker interfaces are : Serializable, Remote, Cloneable
What are checked and unchecked exceptions?
Checked exceptions are the ones which you expect beforehand to be raised when an exceptional condition occurs and so write your code in a try-catch block to handle that sufficiently. For example: InsuffucientBalanceException which might be raised when money is being withdrawn from a bank account and the account has insufficient balance. Checked exceptions are sub classes of Exception.
Unchecked exceptions are the ones which cannot be handled in the code. These are rather unexpected exceptions like NullPointerException, OutOfMemoryError, DivideByZeroException, typically, programming errors. Unchecked exceptions are subclasses of RunTimeExceptions.

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

8. What is singleton class?where is it used?
Singleton is a design pattern meant to provide one and only one instance of an object. Other objects can get a reference to this instance through a static method (class constructor is kept private). Why do we need one? Sometimes it is necessary, and often sufficient, to create a single instance of a given class. This has advantages in memory management, and for Java, in garbage collection. Moreover, restricting the number of instances may be necessary or desirable for technological or business reasons--for example, we may only want a single instance of a pool of database connections.
9. What is a compilation unit?
The smallest unit of source code that can be compiled, i.e. a .java file.
10. Is string a wrapper class?
String is a class, but not a wrapper class. Wrapper classes like (Integer) exist for each primitive type. They can be used to convert a primitive data value into an object, and vice-versa.

11. Why java does not have multiple inheritance?

The Java design team strove to make Java:

- Simple, object oriented, and familiar
- Robust and secure
- Architecture neutral and portable
- High performance
- Interpreted, threaded, and dynamic

The reasons for omitting multiple inheritance from the Java language mostly stem from the "simple, object oriented, and familiar" goal. As a simple language, Java's creators wanted a language that most developers could grasp without extensive training. To that end, they worked to make the language as similar to C++ as possible (familiar) without carrying over C++'s unnecessary complexity (simple).

In the designers' opinion, multiple inheritance causes more problems and confusion than it solves. So they cut multiple inheritance from the language (just as they cut operator overloading). The designers' extensive C++ experience taught them that multiple inheritance just wasn't worth the headache.

12. What is a resource bundle?
In its simplest form, a resource bundle is represented by a text file containing keys and a text value for each key.
13. What is transient variable?
Transient variable can't be serialize. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStream the value of the variable becomes null.
14. What is Collection API?
The Collection API is a set of classes and interfaces that support operation on collections of objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.
Example of classes: `HashSet`, `HashMap`, `ArrayList`, `LinkedList`, `TreeSet` and `TreeMap`.
Example of interfaces: `Collection`, `Set`, `List` and `Map`.
16. Is Iterator a Class or Interface? What is its use?
Iterator is an interface which is used to step through the elements of a Collection.

17. What is similarities/difference between an Abstract class and Interface?
Differences are as follows:

- Interfaces provide a form of multiple inheritance. A class can extend only one other class.
- Interfaces are limited to public methods and constants with no implementation. Abstract classes can have a partial implementation, protected parts, static methods, etc.
- A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.
- Interfaces are slow as it requires extra indirection to to find corresponding method in in the actual class. Abstract classes are fast.

Similarities:

- Neither Abstract classes or Interface can be instantiated.

18. Which containers use a border Layout as their default layout?
The window, Frame and Dialog classes use a border layout as their default layout.
19. 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.

21. 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.
22. 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..
23. 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.
24. Is null a keyword?
The null value is not a keyword.
25. 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.
26. What method is used to specify a container's layout?
The setLayout() method is used to specify a container's layout.
27. Which containers use a FlowLayout as their default layout?
The Panel and Applet classes use the FlowLayout as their default layout.
28. What state does a thread enter when it terminates its processing?
When a thread terminates its processing, it enters the dead state.
29. What is the Collections API?
The Collections API is a set of classes and interfaces that support operations on collections of objects.
30. 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.
31. What is the List interface?
The List interface provides support for ordered collections of objects.
32. 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.
33. What is the Vector class?
The Vector class provides the capability to implement a growable array of objects
34. 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.
35. What is an Iterator interface?
The Iterator interface is used to step through the elements of a Collection.
36. 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.
37. Which method of the Component class is used to set the position and size of a component?
setBounds()
38. 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.
39. 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.
40. Which java.util classes and interfaces support event handling?
The EventObject class and the EventListener interface support event processing.

41. Is sizeof a keyword?
The sizeof operator is not a keyword.
42. What are wrapped classes?
Wrapped classes are classes that allow primitive types to be accessed as objects.
43. 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
44. 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).
45. 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.
47. 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 actors.
49. 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.
50. What is the immediate superclass of the Dialog class? Ans: Window
51. What is clipping?
Ans: Clipping is the process of confining paint operations to a limited area or shape.
52. What is a native method?
A native method is a method that is implemented in a language other than Java.
53. Can a for statement loop indefinitely?
Yes, a for statement can loop indefinitely. For example, consider the following: for(;;) ;
54. What are order of precedence and associativity, and how are they used?
Order of precedence determines the order in which operators are evaluated in expressions. Associatity determines whether an expression is evaluated left-to-right or right-to-left
55. When a thread blocks on I/O, what state does it enter?
A thread enters the waiting state when it blocks on I/O.
56. To what value is a variable of the String type automatically initialized?
The default value of an String type is null.
57. 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.
58. 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.
59. 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.
60. 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.
61. 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.
62. 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.
63. What is the range of the short type?
The range of the short type is -(2^15) to 2^15 - 1.

64. What is the range of the char type?
The range of the char type is 0 to 2^16 - 1.
65. 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.
66. What is the immediate superclass of Menu? MenuItem
67. 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.
68. Which class is the immediate superclass of the MenuComponent class. Object
69. 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.
70. 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.

72. What is the GregorianCalendar class?
The GregorianCalendar provides support for traditional Western calendars.
74. What is the purpose of the Runtime class?
The purpose of the Runtime class is to provide access to the Java runtime system.
75. 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.
76. 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.
77. What is the argument type of a program's main() method?
A program's main() method takes an argument of the String[] type.
78. Which Java operator is right associative? The = operator is right associative.
79. 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.
80. Can a double value be cast to a byte?
Yes, a double value can be cast to a byte.
81. 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.
82. 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.
83. 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.
84. Name two subclasses of the TextComponent class.
TextField and TextArea

85. 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.
87. How are commas used in the intialization and iteration parts of a for statement?
Commas are used to separate multiple statements within the initialization and iteration parts of a for statement.
88. 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..
90. 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.
91. 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.
92. What are the high-level thread states?
The high-level thread states are ready, running, waiting, and dead.
93. 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.
94. Can a Byte object be cast to a double value?
No, an object cannot be cast to a primitive value.
95. 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.
96. What is the difference between the String and StringBuffer classes?
String objects are constants. StringBuffer objects are not.
97. 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.
98. 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.
99. What is the Dictionary class?
The Dictionary class provides the capability to store key-value pairs.
100. 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.
101. 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.

102. 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.

103. 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.

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

105. 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..

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

107. 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.

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

109. 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.

110. 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.

111. 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.

112. 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.

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

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

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

116. 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.

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

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

118. 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.

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

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

121. 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.

121. 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.

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

123. What is the relationship between an event-listener interface and an event-adapter class?
An event-listener interface defines the methods that must be implemented by an event handler for a particular kind of event. An event adapter provides a default implementation of an event-listener interface.

124. 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.

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

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

127. 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.

128. What event results from the clicking of a button?

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

129. 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.

130. 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.

131. 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.

132. What advantage do Java's layout managers provide over traditional windowing systems?
Java uses layout managers to lay out components in a consistent manner across all windowing platforms. Since Java's layout managers aren't tied to absolute sizing and positioning, they are able to accomodate platform-specific differences among windowing systems.

133. 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.

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

135. 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.

136. 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.

137. 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.

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

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

140. 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.

141. 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.

142. 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.

143. Which class is the immediate superclass of the Container class? Ans: Component

144. 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.

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

146. 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

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

148. 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.

149. 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.

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

151. Name four Container classes. Ans: Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

152. 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.

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

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

155. 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.

156. 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.

157. 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.

158. 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.

159. What is the difference between the JDK 1.02 event model and the event-delegation model introduced with JDK 1.1?
The JDK 1.02 event model uses an event inheritance or bubbling approach. In this model, components are required to handle their own events. If they do not handle a particular event, the event is inherited by (or bubbled up to) the component's container. The container then either handles the event or it is bubbled up to its container and so on, until the highest-level container has been tried.. In the event-delegation model, specific objects are designated as event handlers for GUI components. These objects implement event-listener interfaces. The event-delegation model is more efficient than the event-inheritance model because it eliminates the processing required to support the bubbling of unhandled events.
146. How is it possible for two String objects with identical values not to be equal under the == operator?
The == operator compares two objects to determine if they are the same object in memory. It is possible for two String objects to have the same value, but located indifferent areas of memory.

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

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

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

163. 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.

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

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

166. 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.

167. 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.

168. 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.

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

170. Are true and false keywords? Ans: The values true and false are not keywords.

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

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

173. 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.

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

175. 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..

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

177. 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.

178. 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.

179. 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.

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

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

182. 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.

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

184. 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.

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

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

187. 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.

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

189. 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.

190. 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.

191. 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.

192. 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.

193. 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.

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

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

196. 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.

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

198. What is a Java package and how is it used?
A Java package is a naming context for classes and interfaces. A package is used to create a separate name space for groups of classes and interfaces. Packages are also used to organize related classes and interfaces into a single API unit and to control accessibility to these classes and interfaces.
186. What modifiers may be used with a top-level class?
A top-level class may be public, abstract, or final.

199. 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..

200. 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.

201. 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.

202. 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.

203. 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.

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

Ans: getLabel() and setLabel()

205. Which Component subclass is used for drawing and painting? Ans: Canvas

206. 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.

207. 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.

208. 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.

209. 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.

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

211. What is the List interface?
The List interface provides support for ordered collections of objects.
212. What do you know about the garbage collector?
Garbage collector is a runtime component of Java which sits on top of the heap: memory area from which Java objects are created and periodically scans it for objects which are eligible to be reclaimed when there are no references to these objects in the program. There is simply no way to force garbage collection, but you can suggest your intention of getting the object garbage collecetd to Java Virtual Machine by calling

System.gc()

213. Garbage collection thread belongs to which priority..min?normal?max?
The Java garbage collection is implemented as a low priority thread

214. Explain garbage collection?
Garbage collection is one of the most important feature of Java. Garbage collection is also called automatic memory management as JVM automatically removes the unused variables/objects (value is null) from the memory. User program cann't directly free the object from memory, instead it is the job of the garbage collector to automatically free the objects that are no longer referenced by a program. Every class inherits finalize() method from java.lang.Object, the finalize() method is called by garbage collector when it determines no more references to the object exists. In Java, it is good idea to explicitly assign null into a variable when no more in use. I Java on calling System.gc() and Runtime.gc(), JVM tries to recycle the unused objects, but there is no guarantee when all the objects will garbage collected.

216. How you can force the garbage collection?
Garbage collection automatic process and can't be forced.

217. 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.

218. 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

219. Can an objectâ€™s finalize() method be invoked while it is reachable?
An objectâ€™s finalize() method cannot be invoked by the garbage collector while the object is still reachable. However, an objectâ€™s finalize() method may be invoked by other objects.

220. 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.
221. Describe what happens when an object is created in Java?
Several things happen in a particular order to ensure the object is constructed properly: Memory is allocated from heap to hold all instance variables and implementation-specific data of the object and its superclasses. Implemenation-specific data includes pointers to class and method data. The instance variables of the objects

are initialized to their default values. The constructor for the most derived class is invoked. The first thing a constructor does is call the consctructor for its superclasses. This process continues until the constrcutor for java.lang.Object is called, as java.lang.Object is the base class for all objects in java. Before the body of the constructor is executed, all instance variable initializers and initialization blocks are executed. Then the body of the constructor is executed. Thus, the constructor for the base class completes first and constructor for the most derived class completes last.

1#Class Variables (Static Variable)

When a number of objects are created from the same class blueprint, they each have their own distinct copies of instance variables. In the case of the Bicycle class, the instance variables are cadence, gear, and speed. Each Bicycle object has its own values for these variables, stored in different memory locations.

Sometimes, you want to have variables that are common to all objects. This is accomplished with the static modifier. Fields that have the static modifier in their declaration are called static fields or class variables. They are associated with the class, rather than with any object. Every instance of the class shares a class variable, which is in one fixed location in memory. Any object can change the value of a class variable, but class variables can also be manipulated without creating an instance of the class.

For example, suppose you want to create a number of Bicycle objects and assign each a serial number, beginning with 1 for the first object. This ID number is unique to each object and is therefore an instance variable. At the same time, you need a field to keep track of how many Bicycle objects have been created so that you know what ID to assign to the next one. Such a field is not related to any individual object, but to the class as a whole. For this you need a class variable, numberOfBicycles, as follows:

```
public class
 Bicycle{

    private int cadence;
    private int gear;
    private int speed;

    // add an instance variable for the object ID
    private int id;

    // add a class variable for the number of Bicycle objects instantiated
    private static int numberOfBicycles = 0;
            ......
}
```

Class variables are referenced by the class name itself, as in

```
Bicycle.numberOfBicycles
```

This makes it clear that they are class variables.

Note: You can also refer to static fields with an object reference like

```
myBike.numberOfBicycles
```

but this is discouraged because it does not make it clear that they are class variables.

You can use the Bicycle constructor to set the id instance variable and increment the numberOfBicycles class variable:

```
public class Bicycle{
    private int cadence;
    private int gear;
    private int speed;
    private int id;
    private static int numberOfBicycles = 0;
    public Bicycle(int startCadence, int startSpeed, int startGear){
        gear = startGear;
        cadence = startCadence;
        speed = startSpeed;
        // increment number of Bicycles and assign ID number
        id = ++numberOfBicycles;
    }
     // new method to return the ID instance variable
    public int getID() {
        return id;
    }            .....
}
```

2# Class Methods

The Java programming language supports static methods as well as static variables. Static methods, which have the static modifier in their declarations, should be invoked with the class name, without the need for creating an instance of the class, as in

```
ClassName.methodName(args)
```

Note: You can also refer to static methods with an object reference like

```
instanceName.methodName(args)
```

but this is discouraged because it does not make it clear that they are class methods.
A common use for static methods is to access static fields. For example, we could add a static method to the Bicycle class to access the numberOfBicycles static field:

```
public static int getNumberOfBicycles() {
    return numberOfBicycles;
}
```

Not all combinations of instance and class variables and methods are allowed:

- Instance methods can access instance variables and instance methods directly.
- Instance methods can access class variables and class methods directly.
- Class methods can access class variables and class methods directly.
- Class methods cannot access instance variables or instance methods directly—they must use an object reference. Also, class methods cannot use the this keyword as there is no instance for this to refer to.

Method Overriding
Method overriding occurs when a class declares a method that has the same type signature as a method declared by one of its superclasses. Recall that a type signature is a combination of a method name and the sequence of its parameter types. When a method in a subclass overrides a method in a superclass, the method in the superclass is hidden relative to the subclass object.
Method overriding is a very important capability because it forms the basis for **run-time polymorphism**. Recall that polymorphism means "one interface, multiple implementations." The signature of the method defines the interface, and each overridden version provides a unique implementation.
As explained, a superclass reference can be used to refer to a subclass object. **The dynamic method dispatch mechanism** in Java selects the appropriate version of an overridden method to execute based on the class of the executing object, not the type of a variable that references that object. Thus, the actual version of an overridden method that is executed is determined at run-time, not compile time.

The Java compiler issues an error message if a method overrides another method but has a different return type.

Modifier	Function			
	Classes	**Method**	**Variables**	**Constructor**
`public`	Class is publicly accessible (the class can be used by any other class)	Method is publicly accessible	Variables are publicly accessible	Any class can use this constructor.
`protected`	Classes can be declared with security so they can be accessed outside their package by using the public statement in the class declaration. If no explicit statement is made at declaration time, the class may be accessed only from within its own package. *A compile error is generated if any other security modifier is used in the declaration, such as **private or protected**, which are reserved for methods.*	Class members are accessible only to methods in that class and subclasses of that class. This means that protected class members have visibility limited to subclasses	Same as methods	Subclasses of this class and classes in the same package can use this constructor.
`Default(no modifier)`		Only classes in the same package can have access to a class's variables and methods	Same as methods	If no access specifier is provided, the constructor has the default access (sometimes called package private). Only classes within the same package as this class can use this constructor.
`private`	Only inner classes can be **private**	Class members are only accessible by the class they are defined in. This means that no other class has access to private class members, even subclasses	Same as methods	Only this class can use this constructor. Making a constructor private, in essence, makes the class final – the class can't be subclassed, except from within. If all constructors within a class are private, the class might contain public class methods (called *factory methods*) that create and initialize an instance of this class. Other classes can use the factory methods to create an instance of this class.
`static`	The ***`static`** keyword can be used for inner classes, but can not be used to regular classes. The static modifier is **only** applicable to so-called *member classes*	#The static modifier specifies that a variable or method is the same for all objects of a particular class. Typically, new variables are allocated for each instance of a class. When a variable is declared as being static, it is only allocated once regardless of how many objects are instantiated. The result is that all instantiated objects share the same instance of the static variable. Similarly, a static method is one whose implementation is exactly the same for all objects of a particular class. This means that static methods only have access to static variables Declares this is a **1#***class variable* rather than an instance variable. You also use static to declare **2#***class methods*.		x
`Final`	The class cannot be subclassed	A variable has a constant value or that a method cannot be overridden in a subclass. A class member is the final version allowed for the class Indicates that the value of this member cannot change. A compile-time constant is defined by using static and final together		
`abstract`	Class cannot be instantiated; An abstract class can only be subclassed. An abstract class can contain abstract methods. In practice, abstract classes usually provide a complete or partial implementation of at least one method	An abstract method has no implementation and must be a member of an abstract class	x	x
`synchronized`	The class can not be `synchronized`. `We can create synchronized block`	The synchronized modifier is used to specify that a method is thread safe		x
`native`	x	The native modifier is used to identify methods that have native implementations. The native modifier informs the Java compiler that a method's implementation is in an external C file	x	x
`transient`	x	X	Marks member variables that should not be serialized. This component is used in object serialization	x
`volatile`	x	X	Prevents the compiler from performing certain optimizations on a member.	x

Controlling Access to Members of a Class

Access Levels

Specifier	Class	Package	Subclass	World
`private`	Y	N	N	N
no specifier	Y	Y	N	N
`protected`	Y	Y	Y	N
`public`	Y	Y	Y	Y

*`static` keyword

In order to understand the use of the `static` keyword in class declaration, we need to understand th
classes: top-level classes and inner classes.

Top-level classes

You declare a top-level class at the top level as a member of a package. Each top-level class correspo
class name.
A top-level class is by definition already top-level, so there is no point in declaring it static; it is an error

Inner classes

You define an inner class within a top-level class. Depending on how it is defined, an inner class can b

1. Anonymous. Anonymous classes are declared and instantiated within the same statement. They d
once.
The following is an example of an anonymous class:

```
okButton.addActionListener( new ActionListener(){
   public void actionPerformed(ActionEvent e){
      dispose();
   }
});
```

Because an anonymous class doesn't have a normal class declaration where it's possible to use static

2. Local. Local classes are the same as local variables, in the sense that they're created and used insi
can be instantiated as many times as you wish within that block. Like local variables, local classes aren
static.

3. **Member**. Member classes are defined within the body of a class. You can use member classes anyw
declare member classes when you want to use variables and methods of the containing class without
The member class is the only class that you can declare static. When you declare a member class, you
context of an object of the outer class in which this member class is declared. If you want to remove th
class.
When you declare a member class with a static modifier, it becomes a nested top-level class and can b

4**. Nested top-level.** A nested top-level class is a member classes with a static modifier. A nested top-
that it is declared within another class or interface. Nested top-level classes are typically used as a con
new package.
If your main class has a few smaller helper classes that can be used outside the class and make sense
them nested top-level classes.