1. What is the most important feature of Java?  
Java is a platform independent language.

2. What do you mean by platform independence?  
Platform independence means that we can write and compile the java code in one platform (eg Windows) and can execute the class in any other supported platform eg (Linux,Solaris,etc).

3. What is a JVM?  
JVM is Java Virtual Machine which is a run time environment for the compiled java class files.

4. Are JVM's platform independent?  
JVM's are not platform independent. JVM's are platform specific run time implementation provided by the vendor.

5. What is the difference between a JDK and a JVM?  
JDK is Java Development Kit which is for development purpose and it includes execution environment also. But JVM is purely a run time environment and hence you will not be able to compile your source files using a JVM.

6. What is a pointer and does Java support pointers?  
Pointer is a reference handle to a memory location. Improper handling of pointers leads to memory leaks and reliability issues hence Java doesn't support the usage of pointers.

7. What is the base class of all classes?  
java.lang.Object

8. Does Java support multiple inheritance?  
Java doesn't support multiple inheritance.

9. Is Java a pure object oriented language?  
Java uses primitive data types and hence is not a pure object oriented language.

10. Are arrays primitive data types?  
In Java, Arrays are objects.

11. What is difference between Path and Classpath?  
Path and Classpath are operating system level environment variales. Path is used define where the system can find the executables(.exe) files and classpath is used to specify the location .class files.

12. What are local variables?  
Local varaiables are those which are declared within a block of code like methods. Local variables should be initialised before accessing them.

13. What are instance variables?  
Instance variables are those which are defined at the class level. Instance variables need not be initialized before using them as they are automatically initialized to their default values.

14. How to define a constant variable in Java?  
The variable should be declared as static and final. So only one copy of the variable exists for all instances of the class and the value can't be changed also.  
static final int PI = 3.14; is an example for constant.

15. Should a main() method be compulsorily declared in all java classes?  
No not required. main() method should be defined only if the source class is a java application.

16. What is the return type of the main() method?  
Main() method doesn't return anything hence declared void.

17. Why is the main() method declared static?  
main() method is called by the JVM even before the instantiation of the class hence it is declared as static.

18. What is the arguement of main() method?  
main() method accepts an array of String object as arguement.

19. Can a main() method be overloaded?  
Yes. You can have any number of main() methods with different method signature and implementation in the class.

20. Can a main() method be declared final?  
Yes. Any inheriting class will not be able to have it's own default main() method.

21. Does the order of public and static declaration matter in main() method?  
No. It doesn't matter but void should always come before main().

22. Can a source file contain more than one class declaration?  
Yes a single source file can contain any number of Class declarations but only one of the class can be declared as public.

23. What is a package?  
Package is a collection of related classes and interfaces. package declaration should be first statement in a java class.

24. Which package is imported by default?  
java.lang package is imported by default even without a package declaration.

25. Can a class declared as private be accessed outside it's package?  
Not possible.

26. Can a class be declared as protected?  
The protected access modifier cannot be applied to class and interfaces. Methods, fields can be declared protected, however methods and fields in a interface cannot be declared protected.

27. What is the access scope of a protected method?  
A protected method can be accessed by the classes within the same package or by the subclasses of the class in any package.

28. What is the purpose of declaring a variable as final?  
A final variable's value can't be changed. final variables should be initialized before using them.

29. What is the impact of declaring a method as final?  
A method declared as final can't be overridden. A sub-class can't have the same method signature with a different implementation.

30. I don't want my class to be inherited by any other class. What should i do?  
You should declared your class as final. But you can't define your class as final, if it is an abstract class. A class declared as final can't be extended by any other class.

31. Can you give few examples of final classes defined in Java API?  
java.lang.String, java.lang.Math are final classes.

32. How is final different from finally and finalize()?  
final is a modifier which can be applied to a class or a method or a variable. final class can't be inherited, final method can't be overridden and final variable can't be changed. finally is an exception handling code section which gets executed whether an exception is raised or not by the try block code segment. finalize() is a method of Object class which will be executed by the JVM just before garbage collecting object to give a final chance for resource releasing activity.

33. Can a class be declared as static?  
We can not declare top level class as static, but only inner class can be declared static.  
public class Test  
{   
 static class InnerClass

{  
 public static void InnerMethod()  
 { System.out.println("Static Inner Class!");

}  
 }

public static void main(String args[])

{

Test.InnerClass.InnerMethod();

}

}

34. When will you define a method as static?  
When a method needs to be accessed even before the creation of the object of the class then we should declare the method as static.

35. What are the restriction imposed on a static method or a static block of code?  
A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

36. I want to print "Hello" even before main() is executed. How will you acheive that?  
Print the statement inside a static block of code. Static blocks get executed when the class gets loaded into the memory and even before the creation of an object. Hence it will be executed before the main() method. And it will be executed only once.  
37. What is the importance of static variable?  
static variables are class level variables where all objects of the class refer to the same variable. If one object changes the value then the change gets reflected in all the objects.

38. Can we declare a static variable inside a method?  
Static varaibles are class level variables and they can't be declared inside a method. If declared, the class will not compile.  
  
39. What is an Abstract Class and what is it's purpose?  
A Class which doesn't provide complete implementation is defined as an abstract class. Abstract classes enforce abstraction.

40. Can a abstract class be declared final?  
Not possible. An abstract class without being inherited is of no use and hence will result in compile time error.

41. What is use of a abstract variable?  
Variables can't be declared as abstract. only classes and methods can be declared as abstract.

42. Can you create an object of an abstract class?  
Not possible. Abstract classes can't be instantiated.

43. Can a abstract class be defined without any abstract methods?  
Yes it's possible. This is basically to avoid instance creation of the class.

44. Class C implements Interface I containing method m1 and m2 declarations. Class C has provided implementation for method m2. Can i create an object of Class C?  
No not possible. Class C should provide implementation for all the methods in the Interface I. Since Class C didn't provide implementation for m1 method, it has to be declared as abstract. Abstract classes can't be instantiated.

45. Can a method inside a Interface be declared as final?  
No not possible. Doing so will result in compilation error. public and abstract are the only applicable modifiers for method declaration in an interface.

46. Can an Interface implement another Interface?  
Intefaces doesn't provide implementation hence a interface cannot implement another interface.  
  
47. Can an Interface extend another Interface?  
Yes an Interface can inherit another Interface, for that matter an Interface can extend more than one Interface.

48. Can a Class extend more than one Class?  
Not possible. A Class can extend only one class but can implement any number of Interfaces.

49. Why is an Interface be able to extend more than one Interface but a Class can't extend more than one Class?  
Basically Java doesn't allow multiple inheritance, so a Class is restricted to extend only one Class. But an Interface is a pure abstraction model and doesn't have inheritance hierarchy like classes(do remember that the base class of all classes is Object). So an Interface is allowed to extend more than one Interface.

50. Can an Interface be final?  
Not possible. Doing so so will result in compilation error.

51. Can a class be defined inside an Interface?  
Yes it's possible.

52. Can an Interface be defined inside a class?  
Yes it's possible.

53. What is a Marker Interface?  
An Interface which doesn't have any declaration inside but still enforces a mechanism.

54. Which object oriented Concept is achieved by using overloading and overriding?  
Polymorphism.

55. Why does Java not support operator overloading?  
Operator overloading makes the code very difficult to read and maintain. To maintain code simplicity, Java doesn't support operator overloading.

56. Can we define private and protected modifiers for variables in interfaces?  
No.

57. What is Externalizable?  
Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInput in)

58. What modifiers are allowed for methods in an Interface?  
Only public and abstract modifiers are allowed for methods in interfaces.

59. What is a local, member and a class variable?  
Variables declared within a method are "local" variables.  
Variables declared within the class i.e not within any methods are "member" variables (global variables).  
Variables declared within the class i.e not within any methods and are defined as "static" are class variables.

60. What is an abstract method?  
An abstract method is a method whose implementation is deferred to a subclass.  
61. 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.

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

63. 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.

64. 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.

65. 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.

66. 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.

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

68. 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

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

70. 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.

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

72. 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.

73. What do you understand by private, protected and public?  
These are accessibility modifiers. Private is the most restrictive, while public is the least restrictive. There is no real difference between protected and the default type (also known as package protected) within the context of the same package, however the protected keyword allows visibility to a derived class in a different package.

74. What is Downcasting ?  
Downcasting is the casting from a general to a more specific type, i.e. casting down the hierarchy

75. 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.

76. 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.

77. 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).

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

79. 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.

80. 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.

81. What is the range of the char type?  
The range of the char type is 0 to 216 - 1 (i.e. 0 to 65535.)

82. What is the range of the short type?  
The range of the short type is -(215) to 215 - 1. (i.e. -32,768 to 32,767)

83. Why isn't there operator overloading?  
Because C++ has proven by example that operator overloading makes code almost impossible to maintain.

84. What does it mean that a method or field is "static"?  
Static variables and methods are instantiated only once per class. In other words they are class variables, not instance variables. If you change the value of a static variable in a particular object, the value of that variable changes for all instances of that class. Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too). That's how library methods like System.out.println() work. out is a static field in the java.lang.System class.

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

86. 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.

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

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

89. 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.

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

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

92. 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.

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

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

95. 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.

96. 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.

97. 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.

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

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

100. What is the diffrence between inner class and nested class?  
When a class is defined within a scope of another class, then it becomes inner class. If the access modifier of the inner class is static, then it becomes nested class.

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

102. 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.

103. 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**.**

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

105. 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.

106. 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.

107. 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.

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

109. 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.

110. What are the practical benefits, if any, of importing a specific class rather than an entire package (e.g. import java.net.\* versus import java.net.Socket)?  
It makes no difference in the generated class files since only the classes that are actually used are referenced by the generated class file. There is another practical benefit to importing single classes, and this arises when two (or more) packages have classes with the same name. Take java.util.Timer and javax.swing.Timer, for example. If I import java.util.\* and javax.swing.\* and then try to use "Timer", I get an error while compiling (the class name is ambiguous between both packages). Let's say what you really wanted was the javax.swing.Timer class, and the only classes you plan on using in java.util are Collection and HashMap. In this case, some people will prefer to import java.util.Collection and import java.util.HashMap instead of importing java.util.\*. This will now allow them to use Timer, Collection, HashMap, and other javax.swing classes without using fully qualified class names in.

111. Can a method be overloaded based on different return type but same argument type ?  
No, because the methods can be called without using their return type in which case there is ambiquity for the compiler.

112. What happens to a static variable that is defined within a method of a class ?  
Can't do it. You'll get a compilation error.

113. How many static initializers can you have ?  
As many as you want, but the static initializers and class variable initializers are executed in textual order and may not refer to class variables declared in the class whose declarations appear textually after the use, even though these class variables are in scope.

114. What is the difference between method overriding and overloading?  
Overriding is a method with the same name and arguments as in a parent, whereas overloading is the same method name but different arguments

115. What is constructor chaining and how is it achieved in Java ?  
A child object constructor always first needs to construct its parent (which in turn calls its parent constructor.). In Java it is done via an implicit call to the no-args constructor as the first statement.

116. 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.

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

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

119. 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.

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

121. To what value is a variable of the String type automatically initialized?  
The default value of an String type is null.

122. 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.

123. 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.

124. What does it mean that a class or member is final?  
A final class cannot be inherited. A final method cannot be overridden in a subclass. A final field cannot be changed after it's initialized, and it must include an initializer statement where it's declared.

125. What does it mean that a method or class is abstract?  
An abstract class cannot be instantiated. Abstract methods may only be included in abstract classes. However, an abstract class is not required to have any abstract methods, though most of them do. Each subclass of an abstract class must override the abstract methods of its superclasses or it also should be declared abstract.

126. What is a transient variable?  
Transient variable is a variable that may not be serialized.

127. 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.

128. 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.

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

1. 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.

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.

2. What is the purpose of garbage collection in Java, and when is it used?  
The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused.A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

3. Describe synchronization in respect to multithreading.  
With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources.Without synchonization, it is possible for one thread to modify a shared variable while another thread is in the process of using or updating same shared variable. This usually leads to significant errors.

4. What is an abstract class?  
Abstract class must be extended/subclassed (to be useful). It serves as a template. A class that is abstract may not be instantiated (ie. you may not call its constructor), abstract class may contain static data.Any class with an abstract method is automatically abstract itself, and must be declared as such. A class may be declared abstract even if it has no abstract methods. This prevents it from being instantiated.

5. What is the difference between an Interface and an Abstract class?  
An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract.An interface has all public members and no implementation. An abstract class is a class which may have the usual flavors of class members (private, protected, etc.), but has some abstract methods.

6. Explain different way of using thread?  
The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance, the only interface can help.

7. What is an Iterator?  
Some of the collection classes provide traversal of their contents via a java.util.Iterator interface. This interface allows you to walk through a collection of objects, operating on each object in turn.  
Remember when using Iterators that they contain a snapshot of the collection at the time the Iterator was obtained; generally it is not advisable to modify the collection itself while traversing an Iterator.

8. State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.

public: Public class is visible in other packages, field is visible everywhere (class must be public too)  
private : Private variables or methods may be used only by an instance of the same class that declares the variable or method, A private feature may only be accessed by the class that owns the feature.

protected : Is available to all classes in the same package and also available to all subclasses of the class that owns the protected feature. This access is provided even to subclasses that reside in a different package from the class that owns the protected feature.

What you get by default ie, without any access modifier (ie, public private or protected). It means that it is visible to all within a particular package.

9. What is static in java?  
Static means one per class, not one for each object no matter how many instance of a class might exist. This means that you can use them without creating an instance of a class.Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object.A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final. However, you can't override a static method with a nonstatic method. In other words, you can't change a static method into an instance method in a subclass.

10. What is final class?  
A final class can't be extended ie., final class may not be subclassed. A final method can't be overridden when its class is inherited. You can't change value of a final variable (is a constant).

11. What if the main() method is declared as private?  
The program compiles properly but at runtime it will give "main() method not public." message.

12. What if the static modifier is removed from the signature of the main() method?  
Program compiles. But at runtime throws an error "NoSuchMethodError".

13. What if I write static public void instead of public static void?  
Program compiles and runs properly.

14. What if I do not provide the String array as the argument to the method?  
Program compiles but throws a runtime error "NoSuchMethodError".

15. What is the first argument of the String array in main() method?  
The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name.

16. If I do not provide any arguments on the command line, then the String array of main() method will be empty or null?  
It is empty. But not null.

17. How can one prove that the array is not null but empty using one line of code?  
Print args.length. It will print 0. That means it is empty. But if it would have been null then it would have thrown a NullPointerException on attempting to print args.length.

18. What environment variables do I need to set on my machine in order to be able to run Java programs?  
CLASSPATH and PATH are the two variables.

19. Can an application have multiple classes having main() method?  
Yes it is possible. While starting the application we mention the class name to be run. The JVM will look for the Main method only in the class whose name you have mentioned.Hence there is not conflict amongst the multiple classes having main() method.

20. Can I have multiple main() methods in the same class?  
No the program fails to compile. The compiler says that the main() method is already defined in the class.

21. Do I need to import java.lang package any time? Why ?  
No. It is by default loaded internally by the JVM.

22. Can I import same package/class twice? Will the JVM load the package twice at runtime?  
One can import the same package or same class multiple times. Neither compiler nor JVM complains about it. And the JVM will internally load the class only once no matter how many times you import the same class.

23. What are Checked and UnChecked Exception?  
A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses. Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown.Example: IOException thrown by java.io.FileInputStream's read() method·Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked. With an unchecked exception, however, the compiler doesn't force client programmers either to catch the exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown.Example: StringIndexOutOfBoundsException thrown by String's charAt() method· Checked exceptions must be caught at compile time. Runtime exceptions do not need to be. Errors often cannot be.  
  
24. What is Overriding?  
When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.

25. Are the imports checked for validity at compile time? Example: will the code containing an import such as java.lang.ABCD compile?  
Yes the imports are checked for the semantic validity at compile time. The code containing above line of import will not compile. It will throw an error saying, can not resolve symbol  
symbol : class ABCD  
location: package io  
import java.io.ABCD;

26. Does importing a package imports the subpackages as well? Example: Does importing com.MyTest.\* also import com.MyTest.UnitTests.\*?  
No you will have to import the subpackages explicitly. Importing com.MyTest.\* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

27. What is the difference between declaring a variable and defining a variable?  
In declaration we just mention the type of the variable and it's name. We do not initialize it. But defining means declaration + initialization.Example: String s; is just a declaration while String s = new String ("abcd"); Or String s = "abcd"; are both definitions.

28. What is the default value of an object reference declared as an instance variable?  
The default value will be null unless we define it explicitly.

29. Can a top level class be private or protected?  
No. A top level class cannot be private or protected. It can have either "public" or no modifier. If it does not have a modifier it is supposed to have a default access.

If a top level class is declared as private the compiler will complain that the "modifier private is not allowed here". This means that a top level class can not be private. Same is the case with protected.

31. Primitive data types are passed by reference or pass by value?  
Primitive data types are passed by value.

32. Objects are passed by value or by reference?   
Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object.

33. What is serialization?   
Serialization is a mechanism by which you can save the state of an object by converting it to a byte stream.

34. How do I serialize an object to a file?  
The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a fileoutputstream. This will save the object to a file.

35. Which methods of Serializable interface should I implement?  
The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods.

36. How can I customize the seralization process? i.e. how can one have a control over the serialization process?  
Yes it is possible to have control over serialization process. The class should implement Externalizable interface. This interface contains two methods namely readExternal and writeExternal.You should implement these methods and write the logic for customizing the serialization process.  
  
37. What is the common usage of serialization?  
Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serilazed.

38. What is Externalizable interface?  
Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism.Thus if your class implements this interface, you can customize the serialization process by implementing these methods.

39. When you serialize an object, what happens to the object references included in the object?  
The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process.Thus when an object is serialized, all the included objects are also serialized alongwith the original obect.

40. What one should take care of while serializing the object?  
One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

41. What happens to the static fields of a class during serialization?  
There are three exceptions in which serialization doesnot necessarily read and write to the stream. These are

1. Serialization ignores static fields, because they are not part of ay particular state state.  
2. Base class fields are only hendled if the base class itself is serializable.  
3. Transient fields.

42. Does Java provide any construct to find out the size of an object?  
No, there is not sizeof operator in Java. So there is not direct way to determine the size of an object directly in Java.

43. What are wrapper classes?  
Java provides specialized classes corresponding to each of the primitive data types. These are called wrapper classes. They are example: Integer, Character, Double etc.

44. Why do we need wrapper classes?  
It is sometimes easier to deal with primitives as objects. Moreover most of the collection classes store objects and not primitive data types. And also the wrapper classes provide many utility methods also.Because of these resons we need wrapper classes. And since we create instances of these classes we can store them in any of the collection classes and pass them around as a collection. Also we can pass them around as method parameters where a method expects an object.

45. What are checked exceptions?  
Checked exception are those which the Java compiler forces you to catch.Example: IOException are checked exceptions.

46. What are runtime exceptions?  
Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic etc. These are not checked by the compiler at compile time.

47. What is the difference between error and an exception?   
An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error.These JVM errors and you can not repair them at runtime. While exceptions are conditions that occur because of bad input etc. Example: FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference.In most of the cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.).

48. How to create custom exceptions?  
Your class should extend class Exception, or some more specific type thereof.49. If I want an object of my class to be thrown as an exception object, what should I do?The class should extend from Exception class. Or you can extend your class from some more precise exception type also.

50. If my class already extends from some other class what should I do if I want an instance of my class to be thrown as an exception object?  
One can not do anytihng in this scenarion. Because Java does not allow multiple inheritance and does not provide any exception interface as well.

51. How does an exception permeate through the code?  
An unhandled exception moves up the method stack in search of a matching When an exception is thrown from a code which is wrapped in a try block followed by one or more catch blocks, a search is made for matching catch block. If a matching type is found then that block will be invoked. If a matching type is not found then the exception moves up the method stack and reaches the caller method. Same procedure is repeated if the caller method is included in a try catch block. This process continues until a catch block handling the appropriate type of exception is found. If it does not find such a block then finally the program terminates.

52. What are the different ways to handle exceptions?  
There are two ways to handle exceptions,1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions. And   
2. List the desired exceptions in the throws clause of the method and let the caller of the method hadle those exceptions.

53. Is it necessary that each try block must be followed by a catch block?  
It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block or a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

54. If I write return at the end of the try block, will the finally block still execute?  
Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.

55. If I write System.exit(0); at the end of the try block, will the finally block still execute?  
No. In this case the finally block will not execute because when you say System.exit(0); the control immediately goes out of the program, and thus finally never executes.

57. 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.

58. 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.

59. 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.

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

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. 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.

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

64. 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.

65. 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.

66. 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.

67. What is daemon thread and which method is used to create the daemon thread?  
Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system.setDaemon method is used to create a daemon thread.

69. What are the steps in the JDBC connection?  
While making a JDBC connection we go through the following steps :  
Step 1 : Register the database driver by using :  
Class.forName(\" driver classs for that specific database\" );  
Step 2 : Now create a database connection using :Connection con = DriverManager.getConnection(url,username,password);  
Step 3: Now Create a query using :  
Statement stmt = Connection.Statement(\"select \* from TABLE NAME\");  
Step 4 : Exceute the query :stmt.exceuteUpdate();

70. 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 exceptionis executed. The remaining catch clauses are ignored.

71. 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.

72. 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.

73. 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.

74. What is Externalizable?  
Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInput in).

75. What modifiers are allowed for methods in an Interface?  
Only public and abstract modifiers are allowed for methods in interfaces.

76. What are some alternatives to inheritance?  
Delegation is an alternative to inheritance. Delegation means that you include an instance of another class as an instance variable, and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn't force you to accept all the methods of the super class: you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to re-use (because it is not a subclass).

77. What does it mean that a method or field is "static"?  
Static variables and methods are instantiated only once per class. In other words they are class variables, not instance variables. If you change the value of a static variable in a particular object, the value of that variable changes for all instances of that class.Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too). That's how library methods like System.out.println() work out is a static field in the java.lang.System class.

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

79. 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.

80. Is Empty .java file a valid source file?  
Yes. An empty .java file is a perfectly valid source file.

81. Can a .java file contain more than one java classes?  
Yes. A .java file contain more than one java classes, provided at the most one of them is a public class.

82. Is String a primitive data type in Java?  
No. String is not a primitive data type in Java, even though it is one of the most extensively used object. Strings in Java are instances of String class defined in java.lang package.

83. Is main a keyword in Java?  
No. main is not a keyword in Java.

84. Is next a keyword in Java?  
No. next is not a keyword.

85. Is delete a keyword in Java?  
No. delete is not a keyword in Java. Java does not make use of explicit destructors the way C++ does.

86. Is exit a keyword in Java?  
No. To exit a program explicitly you use exit method in System object.

87. What happens if you dont initialize an instance variable of any of the primitive types in Java?  
Java by default initializes it to the default value for that primitive type. Thus an int will be initialized to 0(zero), a boolean will be initialized to false.

88. What will be the initial value of an object reference which is defined as an instance variable?  
The object references are all initialized to null in Java. However in order to do anything useful with these references, you must set them to a valid object, else you will get NullPointerExceptions everywhere you try to use such default initialized references.

89. What are the different scopes for Java variables?  
The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point in time.

1. Instance : - These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object accessible.

2. Local : - These are the variables that are defined within a method. They remain accessbile only during the course of method excecution. When the method finishes execution, these variables fall out of scope.

3. Static: - These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance.

90. What is the default value of the local variables?  
The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the java compiler will not compile the code. It will complain abt the local varaible not being initilized.

91. How many objects are created in the following piece of code?  
MyClass c1, c2, c3; c1 = new MyClass (); c3 = new MyClass ();  
Only 2 objects are created, c1 and c3. The reference c2 is only declared and not initialized.

92. Can a public class MyClass be defined in a source file named YourClass.java?  
No. The source file name, if it contains a public class, must be the same as the public class name itself with a .java extension.

93. Can main() method be declared final?  
Yes, the main() method can be declared final, in addition to being public static.

94. What is HashMap and Map?  
Map is an Interface and Hashmap is the class that implements Map.

95. Difference between HashMap and HashTable?  
The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls. (HashMap allows null values as key and value whereas Hashtable doesnt allow).HashMap does not guarantee that the order of the map will remain constant over time. HashMap is unsynchronized and Hashtable is synchronized.

96. Difference between Vector and ArrayList?  
Vector is synchronized whereas arraylist is not.

98. What will be the default values of all the elements of an array defined as an instance variable?  
If the array is an array of primitive types, then all the elements of the array will be initialized to the default value corresponding to that primitive type.Example: All the elements of an array of int will be initialized to 0(zero), while that of boolean type will be initialized to false. Whereas if the array is an array of references (of any type), all the elements will be initialized to null.

1.what is a transient variable?   
A transient variable is a variable that may not be serialized.

3.Why do threads block on I/O?   
Threads block on i/o (that is enters the waiting state) so that other threads may execute while the i/o Operation is performed.

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

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

7. What's new with the stop(), suspend() and resume() methods in JDK 1.2?   
The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

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

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

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

12. What state does a thread enter when it terminates its processing?   
When a thread terminates its processing, it enters the dead state.   
  
13. What is the Collections API?   
The Collections API is a set of classes and interfaces that support operations on collections of objects.

14. Which characters may be used as the second character of an identifier, but not as the first character of an identifier?   
The digits 0 through 9 may not be used as the first character of an identifier but they may be used after the first character of an identifier.

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

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

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

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

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

20. What is the difference between the >> and >>> operators?   
The >> operator carries the sign bit when shifting right. The >>> zero-fills bits that have been shifted out.   
  
22. How many bits are used to represent Unicode, ASCII, UTF16, and UTF-8 characters?   
Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

23.What is the difference between yielding and sleeping?   
When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

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

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

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

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

28. What restrictions are placed on the location of a package statement within a source code file?   
A package statement must appear as the first line in a source code file (excluding blank lines and comments).

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

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

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

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

38. What are order of precedence and associativity, and how are they used?   
Order of precedence determines the order in which operators are evaluated in expressions. Associatity determines whether an expression is evaluated left-to-right or right-to-left

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

40. To what value is a variable of the String type automatically initialized?   
The default value of an String type is null.

41. What is the catch or declare rule for method declarations?   
If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

71. 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

93. What happens when a thread cannot acquire a lock on an object?   
If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

166. What is the purpose of garbage collection?   
The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources may be reclaimed and reused.   
  
167. What is a compilation unit?   
A compilation unit is a Java source code file.   
  
169. What restrictions are placed on method overriding?   
Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden method.

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

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

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

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

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

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

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

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

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

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

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

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

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

185. What is a Java package and how is it used?   
A Java package is a naming context for classes and interfaces. A package is used to create a separate name space for groups of classes and interfaces. Packages are also used to organize related classes and interfaces into a single API unit and to control accessibility to these classes and interfaces.

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

187. What are the Object and Class classes used for?   
The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program.   
188. How does a try statement determine which catch clause should be used to handle an exception?   
When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

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

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

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

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

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

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

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

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