Introduction to Java Programming

1) The Java interpreter is used for the execution of the source code.
a) True
b) False
Ans: a.

2) On successful compilation a file with the class extension is created.
a) True
b) False
Ans: a.

3) The Java source code can be created in a Notepad editor.
a) True
b) False
Ans: a.

4) The Java Program is enclosed in a class definition.
a) True
b) False
Ans: a.

5) What declarations are required for every Java application?
Ans: A class and the main( ) method declarations.

6) What are the two parts in executing a Java program and their purposes?
Ans: Two parts in executing a Java program are:
Java Compiler and Java Interpreter.
The Java Compiler is used for compilation and the Java Interpreter is used for execution of the application.

7) What are the three OOPs principles and define them?
Ans : Encapsulation, Inheritance and Polymorphism are the three OOPs Principles.

Encapsulation:
Is the Mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse.

Inheritance:
Is the process by which one object acquires the properties of another object.

Polymorphism:
Is a feature that allows one interface to be used for a general class of actions.

8) What is a compilation unit?
Ans : Java source code file.

9) What output is displayed as the result of executing the following statement?
System.out.println("// Looks like a comment.");
// Looks like a comment
The statement results in a compilation error
Looks like a comment
No output is displayed
Ans : a.

10) In order for a source code file, containing the public class Test, to successfully compile, which of the following must be true?
It must have a package statement
It must be named Test.java
It must import java.lang
It must declare a public class named Test
Ans : b

11) What are identifiers and what is naming convention?

Ans : Identifiers are used for class names, method names and variable names. An identifier may be any descriptive sequence of upper case & lower case letters,numbers or underscore or dollar sign and must not begin with numbers.

12) What is the return type of program's main( ) method?

Ans : void

13) What is the argument type of program's main( ) method?

Ans : string array.

14) Which characters are as first characters of an identifier?

Ans : A – Z, a – z, _ ,$

15) What are different comments?

Ans : 1) // -- single line comment
2) /* --
*/ multiple line comment
3) /** --
*/ documentation

16) What is the difference between constructor method and method?

Ans : Constructor will be automatically invoked when an object is created. Whereas method has to be call explicitly.

17) What is the use of bin and lib in JDK?

Ans : Bin contains all tools such as javac, applet viewer, awt tool etc., whereas Lib contains all packages and variables.

Data types,variables and Arrays

1) What is meant by variable?

Ans: Variables are locations in memory that can hold values. Before assigning any value to a variable, it must be declared.

2) What are the kinds of variables in Java? What are their uses?

Ans: Java has three kinds of variables namely, the instance variable, the local variable and the class variable.

Local variables are used inside blocks as counters or in methods as temporary variables and are used to store information needed by a single method.

Instance variables are used to define attributes or the state of a particular object and are used to store information needed by multiple methods in the objects.

Class variables are global to a class and to all the instances of the class and are useful for communicating between different objects of all the same class or keeping track of global states.

3) How are the variables declared?

Ans: Variables can be declared anywhere in the method definition and can be initialized during their declaration.They are commonly declared before usage at the beginning of the definition.

Variables with the same data type can be declared together. Local variables must be given a value before usage.

4) What are variable types?

Ans: Variable types can be any data type that java supports, which includes the eight primitive data types, the name of a class or interface and an array.

5) How do you assign values to variables?

Ans: Values are assigned to variables using the assignment operator =.

6) What is a literal? How many types of literals are there?

Ans: A literal represents a value of a certain type where the type describes how that value behaves.

There are different types of literals namely number literals, character literals, boolean literals, string literals,etc.

7) What is an array?

Ans: An array is an object that stores a list of items.

8) How do you declare an array?
Ans: Array variable indicates the type of object that the array holds.
Ex: int arr[];
9) Java supports multidimensional arrays.
a)True
b)False
Ans: a.
10) An array of arrays can be created.
a)True
b)False
Ans: a.
11) What is a string?
Ans: A combination of characters is called as string.
12) Strings are instances of the class String.
a)True
b)False
Ans: a.
13) When a string literal is used in the program, Java automatically creates instances of the string class.
a)True
b)False
Ans: a.
14) Which operator is to create and concatenate string?
Ans: Addition operator(+).
15) Which of the following declare an array of string objects?
String[ ] s;
String [ ]s:
String[s]:
String s[ ]:
Ans : a, b and d
16) What is the value of a[3] as the result of the following array declaration?
1
2
3
4
Ans : d
17) Which of the following are primitive types?
byte
String
integer
Float
Ans : a.
18) What is the range of the char type?
0 to 2^{16}
0 to 2^{15}
0 to 2^{16}-1
0 to 2^{15}-1
Ans. d
19) What are primitive data types?
Ans : byte, short, int, long
float, double
boolean
char
20) What are default values of different primitive types?
Ans : int - 0

short - 0
byte - 0
long - 0 l
float - 0.0 f
double - 0.0 d
boolean - false
char - null

21) Converting of primitive types to objects can be explicitly.
a)True
b)False
Ans: b.

22) How do we change the values of the elements of the array?
Ans : The array subscript expression can be used to change the values of the elements of the array.

23) What is final varaible?
Ans : If a variable is declared as final variable, then you can not change its value. It becomes constant.

24) What is static variable?
Ans : Static variables are shared by all instances of a class.

Operators

1) What are operators and what are the various types of operators available in Java?
Ans: Operators are special symbols used in expressions.
The following are the types of operators:
Arithmetic operators,
Assignment operators,
Increment & Decrement operators,
Logical operators,
Biwise operators,
Comparison/Relational operators and
Conditional operators

2) The ++ operator is used for incrementing and the -- operator is used for decrementing.
a)True
b)False
Ans: a.

3) Comparison/Logical operators are used for testing and magnitude.
a)True
b)False
Ans: a.

4) Character literals are stored as unicode characters.
a)True
b)False
Ans: a.

5) What are the Logical operators?
Ans: OR(|), AND(&), XOR(^) AND NOT(~).

6) What is the % operator?
Ans : % operator is the modulo operator or reminder operator. It returns the reminder of dividing the first operand by second operand.

7) What is the value of 111 % 13?
3
5
7

9

Ans : c.

8) Is &&= a valid operator?

Ans : No.

9) Can a double value be cast to a byte?

Ans : Yes

10) Can a byte object be cast to a double value ?

Ans : No. An object cannot be cast to a primitive value.

11) What are order of precedence and associativity?

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

12) Which Java operator is right associativity?

Ans : = operator.

13) What is the difference between prefix and postfix of -- and ++ operators?

Ans : The prefix form returns the increment or decrement operation and returns the value of the increment or decrement operation.

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

14) What is the result of expression 5.45 + "3,2"?

The double value 8.6

The string ""8.6"

The long value 8.

The String "5.453.2"

Ans : d

15) What are the values of x and y ?

x = 5; y = ++x;

Ans : x = 6; y = 6

16) What are the values of x and z?

x = 5; z = x++;

Ans : x = 6; z = 5

Control Statements

1) What are the programming constructs?

Ans: a) Sequential

b) Selection -- if and switch statements

c) Iteration -- for loop, while loop and do-while loop

```
2) class conditional {
public static void main(String args[]) {
int i = 20;
int j = 55;
int z = 0;
z = i < j ? i : j; // ternary operator
System.out.println("The value assigned is " + z);
}
}
```

What is output of the above program?

Ans: The value assigned is 20

3) The switch statement does not require a break.

a)True

b)False

Ans: b.
4) The conditional operator is otherwise known as the ternary operator.
a)True
b)False
Ans: a.
5) The while loop repeats a set of code while the condition is false.
a)True
b)False
Ans: b.
6) The do-while loop repeats a set of code atleast once before the condition is tested.
a)True
b)False
Ans: a.
7) What are difference between break and continue?
Ans: The break keyword halts the execution of the current loop and forces control out of the loop.
The continue is similar to break, except that instead of halting the execution of the loop, it starts the next iteration.

8) The for loop repeats a set of statements a certain number of times until a condition is matched.
a)True
b)False
Ans: a.
9) Can a for statement loop indefintely?
Ans : Yes.
10) What is the difference between while statement and a do statement/
Ans : 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.

Introduction to Classes and Methods

1) Which is used to get the value of the instance variables?
Ans: Dot notation.
2) The new operator creates a single instance named class and returns a reference to that object.
a)True
b)False
Ans: a.
3) A class is a template for multiple objects with similar features.
a)True
b)False
Ans: a.
4) What is mean by garbage collection?
Ans: When an object is no longer referred to by any variable, Java automatically reclaims memory used by that object. This is known as garbage collection.
5) What are methods and how are they defined?
Ans: Methods are functions that operate on instances of classes in which they are defined.Objects can communicate with each other using methods and can call methods in other classes.
Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method.
A method's signature is a combination of the first three parts mentioned above.

6) What is calling method?
Ans: Calling methods are similar to calling or referring to an instance variable. These methods are accessed using dot notation.
Ex: obj.methodname(param1,param2)
7) Which method is used to determine the class of an object?
Ans: getClass( ) method can be used to find out what class the belongs to. This class is defined in the object class and is available to all objects.
8) All the classes in java.lang package are automatically imported when a program is compiled.
a)True
b)False
Ans: a.
9) How can class be imported to a program?
Ans: To import a class, the import keyword should be used as shown.;
import classname;
10) How can class be imported from a package to a program?
Ans: import java . packagename . classname (or) import java.package name.*;
11) What is a constructor?
Ans: A constructor is a special kind of method that determines how an object is initialized when created.
12) Which keyword is used to create an instance of a class?
Ans: new.
13) Which method is used to garbage collect an object?
Ans: finalize ().
14) Constructors can be overloaded like regular methods.
a)True
b)False
Ans: a.
15) What is casting?
Ans: Casting is bused to convert the value of one type to another.

16) Casting between primitive types allows conversion of one primitive type to another.
a)True
b)False
Ans: a.
17) Casting occurs commonly between numeric types.
a)True
b)False
Ans: a.
18) Boolean values can be cast into any other primitive type.
a)True
b)False
Ans: b.
19) Casting does not affect the original object or value.
a)True
b)False
Ans: a.
20) Which cast must be used to convert a larger value into a smaller one?
Ans: Explicit cast.
21) Which cast must be used to cast an object to another class?
Ans: Specific cast.
22) Which of the following features are common to both Java & C++?

A.The class declaration
b.The access modifiers
c.The encapsulation of data & methods with in objects
d.The use of pointers
Ans: a,b,c.
23) Which of the following statements accurately describe the use of access modifiers within a class definition?
a.They can be applied to both data & methods
b.They must precede a class's data variables or methods
c.They can follow a class's data variables or methods
d.They can appear in any order
e.They must be applied to data variables first and then to methods
Ans: a,b,d.
24) Suppose a given instance variable has been declared private.
Can this instance variable be manipulated by methods out side its class?
a.yes
b.no
Ans: b.
25) Which of the following statements can be used to describe a public method?
a.It is accessible to all other classes in the hierarchy
b.It is accessabldе only to subclasses of its parent class
c.It represents the public interface of its class
d.The only way to gain access to this method is by calling one of the public class methods
Ans: a,c.
26) Which of the following types of class members can be part of the internal part of a class?
a.Public instance variables
b.Private instance variables
c.Public methods
d.Private methods
Ans: b,d.
27) You would use the ____ operator to create a single instance of a named class.
a.new
b.dot
Ans: a.
28) Which of the following statements correctly describes the relation between an object and the instance variable it stores?
a.Each new object has its own distinctive set of instance variables
b.Each object has a copy of the instance variables of its class
c.the instance variable of each object are seperate from the variables of other objects
d.The instance variables of each object are stored together with the variables of other objects
Ans: a,b,c.
29) If no input parameters are specified in a method declaration then the declaration will include __.
a.an empty set of parantheses
b.the term void
Ans: a.
30) What are the functions of the dot(.) operator?
a.It enables you to access instance variables of any objects within a class
b.It enables you to store values in instance variables of an object
c.It is used to call object methods
d.It is to create a new object
Ans: a,b,c.

31) Which of the following can be referenced by this variable?
a.The instance variables of a class only
b.The methods of a class only
c.The instance variables and methods of a class
Ans: c.
32) The this reference is used in conjunction with ___methods.
a.static
b.non-static
Ans: b.
33) Which of the following operators are used in conjunction with the this and super references?
a.The new operator
b.The instanceof operator
c.The dot operator
Ans: c.
34) A constructor is automatically called when an object is instantiated
a. true
b. false
Ans: a.
35) When may a constructor be called without specifying arguments?
a. When the default constructor is not called
b. When the name of the constructor differs from that of the class
c. When there are no constructors for the class
Ans: c.
36) Each class in java can have a finalizer method
a. true
b.false
Ans: a.
37) When an object is referenced, does this mean that it has been identified by the finalizer method for garbage collection?
a.yes
b.no
Ans: b.
38) Because finalize () belongs to the java.lang.Object class, it is present in all ___.
a.objects
b.classes
c.methods
Ans: b.
39) Identify the true statements about finalization.
a.A class may have only one finalize method
b.Finalizers are mostly used with simple classes
c.Finalizer overloading is not allowed
Ans: a,c.
40) When you write finalize() method for your class, you are overriding a finalizer inherited from a super class.
a.true
b.false
Ans: a.
41) Java memory management mechanism garbage collects objects which are no longer referenced
a true
b.false
Ans: a.
42) are objects referenced by a variable candidates for garbage collection when the variable goes out of scope?

a yes
b. no
Ans: a.
43) Java's garbage collector runs as a ___ priority thread waiting for __priority threads to relinquish the processor.
a.high
b.low
Ans: a,b.
44) The garbage collector will run immediately when the system is out of memory
a.true
b.false
Ans: a.
45) You can explicitly drop a object reference by setting the value of a variable whose data type is a reference type to ___
Ans: null
46) When might your program wish to run the garbage collecter?
a. before it enters a compute-intense section of code
b. before it enters a memory-intense section of code
c. before objects are finalized
d. when it knows there will be some idle time
Ans: a,b,d
47) For externalizable objects the class is solely responsible for the external format of its contents
a.true
b.false
Ans: a
48) When an object is stored, are all of the objects that are reachable from that object stored as well?
a.true
b.false
Ans: a
49) The default__ of objects protects private and trancient data, and supports the __ of the classes
a.evolution
b.encoding
Ans: b,a.
50) Which are keywords in Java?
a) NULL
b) sizeof
c) friend
d) extends
e) synchronized
Ans : d and e
51) When must the main class and the file name coincide?
Ans :When class is declared public.
52) What are different modifiers?
Ans : public, private, protected, default, static, trancient, volatile, final, abstract.
53) What are access modifiers?
Ans : public, private, protected, default.
54) What is meant by "Passing by value" and " Passing by reference"?
Ans : objects – pass by referrence
Methods - pass by value
55) Is a class a subclass of itself?
Ans : A class is a subclass itself.

56) What modifiers may be used with top-level class?
Ans : public, abstract, final.
57) What is an example of polymorphism?
Inner class
Anonymous classes
Method overloading
Method overriding
Ans : c

Packages and interface
1) What are packages ? what is use of packages ?
Ans :The package statement defines a name space in which classes are stored.If you omit the package, the classes are put into the default package.
Signature... package pkg;
Use: * It specifies to which package the classes defined in a file belongs to. * Package is both naming and a visibility control mechanism.
2) What is difference between importing "java.applet.Applet" and "java.applet.*;" ?
Ans :"java.applet.Applet" will import only the class Applet from the package java.applet
Where as "java.applet.*" will import all the classes from java.applet package.
3) What do you understand by package access specifier?
Ans : public: Anything declared as public can be accessed from anywhere
private: Anything declared in the private can't be seen outside of its class.
default: It is visible to subclasses as well as to other classes in the same package.
4) What is interface? What is use of interface?
Ans : It is similar to class which may contain method's signature only but not bodies. Methods declared in interface are abstract methods. We can implement many interfaces on a class which support the multiple inheritance.
5) Is it is necessary to implement all methods in an interface?
Ans : Yes. All the methods have to be implemented.
6) Which is the default access modifier for an interface method?
Ans : public.
7) Can we define a variable in an interface ?and what type it should be ?
Ans : Yes we can define a variable in an interface. They are implicitly final and static.
8) What is difference between interface and an abstract class?
Ans : All the methods declared inside an Interface are abstract. Where as abstract class must have at least one abstract method and others may be concrete or abstract.
In Interface we need not use the keyword abstract for the methods.
9) By default, all program import the java.lang package.
True/False
Ans : True
10) Java compiler stores the .class files in the path specified in CLASSPATH environmental variable.
True/False
Ans : False

11) User-defined package can also be imported just like the standard packages.
True/False
Ans : True
12) When a program does not want to handle exception, the ______class is used.
Ans : Throws
13) The main subclass of the Exception class is _______ class.
Ans : RuntimeException

14) Only subclasses of ______class may be caught or thrown.
Ans : Throwable
15) Any user-defined exception class is a subclass of the _____ class.
Ans : Exception
16) The catch clause of the user-defined exception class should ______ its
Base class catch clause.
Ans : Exception
17) A _______ is used to separate the hierarchy of the class while declaring an
Import statement.
Ans : Package

18) All standard classes of Java are included within a package called _____.
Ans : java.lang
19) All the classes in a package can be simultaneously imported using ____.
Ans : *
20) Can you define a variable inside an Interface. If no, why? If yes, how?
Ans.: YES. final and static
21) How many concrete classes can you have inside an interface?
Ans.: None
22) Can you extend an interface?
Ans.: Yes
23) Is it necessary to implement all the methods of an interface while implementing
the interface?
Ans.: No
24) If you do not implement all the methods of an interface while implementing ,
what specifier should you use for the class ?
Ans.: abstract
25) How do you achieve multiple inheritance in Java?
Ans: Using interfaces.
26) How to declare an interface example?
Ans : access class classname implements interface.
27) Can you achieve multiple interface through interface?
a)True
b) false
Ans : a.
28) Can variables be declared in an interface ? If so, what are the modifiers?
Ans : Yes. final and static are the modifiers can be declared in an interface.
29) What are the possible access modifiers when implementing interface methods?
Ans : public.
30) Can anonymous classes be implemented an interface?
Ans : Yes.
31) Interfaces can’t be extended.
a)True
b)False
Ans : b.
32) Name interfaces without a method?
Ans : Serializable, Cloneble & Remote.
33) Is it possible to use few methods of an interface in a class ? If so, how?
Ans : Yes. Declare the class as abstract.

Exception Handling

1) What is the difference between ‘throw’ and ‘throws’ ?And it’s application?
Ans : Exceptions that are thrown by java runtime systems can be handled by Try and catch blocks. With throw exception we can handle the exceptions thrown by the program itself. If a method is capable of causing an exception that it does not

handle, it must specify this behavior so the callers of the method can guard against that exception.

2) What is the difference between ‘Exception’ and ‘error’ in java?

Ans : Exception and Error are the subclasses of the Throwable class. Exception class is used for exceptional conditions that user program should catch. With exception class we can subclass to create our own custom exception.

Error defines exceptions that are not excepted to be caught by you program. Example is Stack Overflow.

3) What is ‘Resource leak’?

Ans : Freeing up other resources that might have been allocated at the beginning of a method.

4)What is the ‘finally’ block?

Ans : Finally block will execute whether or not an exception is thrown. If an exception is thrown, the finally block will execute even if no catch statement match the exception. Any time a method is about to return to the caller from inside try/catch block, via an uncaught exception or an explicit return statement, the finally clause is also execute.

5) Can we have catch block with out try block? If so when?

Ans : No. Try/Catch or Try/finally form a unit.

6) What is the difference between the following statements?

Catch (Exception e),
Catch (Error err),
Catch (Throwable t)

Ans :

7) What will happen to the Exception object after exception handling?

Ans : It will go for Garbage Collector. And frees the memory.

8) How many Exceptions we can define in ‘throws’ clause?

Ans : We can define multiple exceptions in throws clause.

Signature is..

type method-name (parameter-list) throws exception-list

9) The finally block is executed when an exception is thrown, even if no catch matches it.

True/False

Ans : True

10) The subclass exception should precede the base class exception when used within the catch clause.

True/False

Ans : True

11) Exceptions can be caught or rethrown to a calling method.

True/False

Ans : True

12) The statements following the throw keyword in a program are not executed.

True/False

Ans : True

13) The toString ( ) method in the user-defined exception class is overridden.

True/False

Ans : True

MULTI THREADING

1) What are the two types of multitasking?

Ans : 1.process-based

2.Thread-based

2) What are the two ways to create the thread?

Ans : 1.by implementing Runnable

2.by extending Thread

3) What is the signature of the constructor of a thread class?

Ans : Thread(Runnable threadob,String threadName)

4) What are all the methods available in the Runnable Interface?

Ans : run()

5) What is the data type for the method isAlive() and this method is available in which class?

Ans : boolean, Thread

6) What are all the methods available in the Thread class?

Ans : 1.isAlive()

2.join()

3.resume()

4.suspend()

5.stop()

6.start()

7.sleep()

8.destroy()

7) What are all the methods used for Inter Thread communication and what is the class in which these methods are defined?

Ans :1. wait(),notify() & notifyall()

2. Object class

8) What is the mechanisam defind by java for the Resources to be used by only one Thread at a time?

Ans : Synchronisation

9) What is the procedure to own the moniter by many threads?

Ans : not possible

10) What is the unit for 1000 in the below statement?

ob.sleep(1000)

Ans : long milliseconds

11) What is the data type for the parameter of the sleep() method?

Ans : long

12) What are all the values for the following level?

max-priority

min-priority

normal-priority

Ans : 10,1,5

13) What is the method available for setting the priority?

Ans : setPriority()

14) What is the default thread at the time of starting the program?

Ans : main thread

15) The word synchronized can be used with only a method.

True/ False
Ans : False
16) Which priority Thread can prompt the lower primary Thread?
Ans : Higher Priority
17) How many threads at a time can access a monitor?
Ans : one
18) What are all the four states associated in the thread?
Ans : 1. new 2. runnable 3. blocked 4. dead
19) The suspend()method is used to teriminate a thread?
True /False
Ans : False
20) The run() method should necessary exists in clases created as subclass of thread?
True /False
Ans : True
21) When two threads are waiting on each other and can't proceed the programe is said to be in a deadlock?
True/False
Ans : True
22) Which method waits for the thread to die ?
Ans : join() method

23) Which of the following is true?
1) wait(),notify(),notifyall() are defined as final & can be called only from with in a synchronized method
2) Among wait(),notify(),notifyall() the wait() method only throws IOException
3) wait(),notify(),notifyall() & sleep() are methods of object class
1
2
3
1 & 2
1,2 & 3
Ans : D
24) Garbage collector thread belongs to which priority?
Ans : low-priority
25) What is meant by timeslicing or time sharing?
Ans : Timeslicing is the method of allocating CPU time to individual threads in a priority schedule.
26) What is meant by daemon thread? In java runtime, what is it's role?
Ans : Daemon thread is a low priority thread which runs intermittently in the background doing the garbage collection operation for the java runtime system.

Inheritance
1) What is the difference between superclass & subclass?
Ans : A super class is a class that is inherited whereas subclass is a class that does the inheriting.
2) Which keyword is used to inherit a class?
Ans : extends
3) Subclasses methods can access superclass members/ attributes at all times?
True/False
Ans : False

4) When can subclasses not access superclass members?
Ans : When superclass is declared as private.
5) Which class does begin Java class hierarchy?
Ans : Object class
6) Object class is a superclass of all other classes?
True/False
Ans : True
7) Java supports multiple inheritance?
True/False
Ans : False
8) What is inheritance?
Ans : Deriving an object from an existing class. In the other words, Inheritance is the process of inheriting all the features from a class
9) What are the advantages of inheritance?
Ans : Reusability of code and accessibility of variables and methods of the superclass by subclasses.
10) Which method is used to call the constructors of the superclass from the subclass?
Ans : super(argument)
11) Which is used to execute any method of the superclass from the subclass?
Ans : super.method-name(arguments)
12) Which methods are used to destroy the objects created by the constructor methods?
Ans : finalize()
13) What are abstract classes?
Ans : Abstract classes are those for which instances can’t be created.
14) What must a class do to implement an interface?
Ans: It must provide all of the methods in the interface and identify the interface in its implements clause.
15) Which methods in the Object class are declared as final?
Ans : getClass(), notify(), notifyAll(), and wait()
16) Final methods can be overridden.
True/False
Ans : False
17) Declaration of methods as final results in faster execution of the program?
True/False
Ans: True
18) Final variables should be declared in the beginning?
True/False
Ans : True
19) Can we declare variable inside a method as final variables? Why?
Ans : Cannot because, local variable cannot be declared as final variables.
20) Can an abstract class may be final?
Ans : An abstract class may not be declared as final.
21) Does a class inherit the constructors of it's super class?
Ans: A class does not inherit constructors from any of it's super classes.
22) What restrictions are placed on method overloading?
Ans: Two methods may not have the same name and argument list but different return types.
23) What restrictions are placed on method overriding?
Ans : Overridden methods must have the same name , argument list , and return type. The overriding method may not limit the access of the method it overridees.The overriding method may not throw any exceptions that may not be thrown by the overridden method.

24) What modifiers may be used with an inner class that is a member of an outer class?
Ans : a (non-local) inner class may be declared as public, protected, private, static, final or abstract.
25) How this() is used with constructors?
Ans: this() is used to invoke a constructor of the same class
26) How super() used with constructors?
Ans : super() is used to invoke a super class constructor
27) Which of the following statements correctly describes an interface?
a)It's a concrete class
b)It's a superclass
c)It's a type of abstract class
Ans: c

28) An interface contains __ methods
a)Non-abstract
b)Implemented
c)unimplemented
Ans:c
STRING HANDLING
Which package does define String and StringBuffer classes?
Ans : java.lang package.
Which method can be used to obtain the length of the String?
Ans : length( ) method.
How do you concatenate Strings?
Ans : By using " + " operator.
Which method can be used to compare two strings for equality?
Ans : equals( ) method.
Which method can be used to perform a comparison between strings that ignores case differences?
Ans : equalsIgnoreCase( ) method.
What is the use of valueOf( ) method?
Ans : valueOf( ) method converts data from its internal format into a human-readable form.
What are the uses of toLowerCase( ) and toUpperCase( ) methods?
Ans : The method toLowerCase( ) converts all the characters in a string from uppercase to
lowercase.
The method toUpperCase( ) converts all the characters in a string from lowercase to uppercase.
Which method can be used to find out the total allocated capacity of a StrinBuffer?
Ans : capacity( ) method.
Which method can be used to set the length of the buffer within a StringBuffer object?
Ans : setLength( ).
What is the difference between String and StringBuffer?
Ans : String objects are constants, whereas StringBuffer objects are not.
String class supports constant strings, whereas StringBuffer class supports growable, modifiable strings.
What are wrapper classes?
Ans : Wrapper classes are classes that allow primitive types to be accessed as objects.
Which of the following is not a wrapper class?
String
Integer

Boolean
Character
Ans : a.

What is the output of the following program?

```
public class Question {
public static void main(String args[]) {
String s1 = "abc";
String s2 = "def";
String s3 = s1.concat(s2.toUpperCase( ) );
System.out.println(s1+s2+s3);
}
}
```

abcdefabcdef
abcabcDEFDEF
abcdefabcDEF
None of the above
ANS : c.

Which of the following methods are methods of the String class?

delete( )
append( )
reverse( )
replace( )
Ans : d.

Which of the following methods cause the String object referenced by s to be changed?

s.concat( )
s.toUpperCase( )
s.replace( )
s.valueOf( )
Ans : a and b.

String is a wrapper class?

True
False
Ans : b.

17) If you run the code below, what gets printed out?

```
String s=new String("Bicycle");

int iBegin=1;

char iEnd=3;

System.out.println(s.substring(iBegin,iEnd));
```

Bic
ic
c) icy
d) error: no method matching substring(int,char)
Ans : b.

18) Given the following declarations

```
String s1=new String("Hello")

String s2=new String("there");

String s3=new String();
```

Which of the following are legal operations?

s3=s1 + s2;

s3=s1 - s2;
c) s3=s1 & s2
d) s3=s1 && s2
Ans : a.

19) Which of the following statements are true?
The String class is implemented as a char array, elements are addressed using the stringname[] convention
b) Strings are a primitive type in Java that overloads the + operator for concatenation

c) Strings are a primitive type in Java and the StringBuffer is used as the matching wrapper type
d) The size of a string can be retrieved using the length property.
Ans : b.

EXPLORING JAVA.LANG

java.lang package is automatically imported into all programs.
True
False
Ans : a

What are the interfaces defined by java.lang?
Ans : Cloneable, Comparable and Runnable.

What are the constants defined by both Flaot and Double classes?
Ans : MAX_VALUE,
MIN_VALUE,
NaN,
POSITIVE_INFINITY,
NEGATIVE_INFINITY and
TYPE.

What are the constants defined by Byte, Short, Integer and Long?
Ans : MAX_VALUE,
MIN_VALUE and
TYPE.

What are the constants defined by both Float and Double classes?
Ans : MAX_RADIX,
MIN_RADIX,
MAX_VALUE,
MIN_VALUE and
TYPE.

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

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

Which class is extended by all other classes?
Ans : Object class is extended by all other classes.

Which class can be used to obtain design information about an object?
Ans : The Class class can be used to obtain information about an object’s design.

Which method is used to calculate the absolute value of a number?
Ans : abs( ) method.

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

Which of the following classes is used to perform basic console I/O?
System

SecurityManager
Math
Runtime
Ans : a.

Which of the following are true?
The Class class is the superclass of the Object class.
The Object class is final.
The Class class can be used to load other classes.
The ClassLoader class can be used to load other classes.
Ans : c and d.

Which of the following methods are methods of the Math class?
absolute( )
log( )
cosine( )
sine( )
Ans : b.

Which of the following are true about the Error and Exception classes?
Both classes extend Throwable.
The Error class is final and the Exception class is not.
The Exception class is final and the Error is not.
Both classes implement Throwable.
Ans : a.

Which of the following are true?
The Void class extends the Class class.
The Float class extends the Double class.
The System class extends the Runtime class.
The Integer class extends the Number class.
Ans : d.

17) Which of the following will output -4.0
System.out.println(Math.floor(-4.7));
System.out.println(Math.round(-4.7));
System.out.println(Math.ceil(-4.7));
d) System.out.println(Math.Min(-4.7));
Ans : c.

18) Which of the following are valid statements
a) public class MyCalc extends Math
b) Math.max(s);
c) Math.round(9.99,1);
d) Math.mod(4,10);
e) None of the above.
Ans : e.

19) What will happen if you attempt to compile and run the following code?

```
Integer ten=new Integer(10);

Long nine=new Long (9);

System.out.println(ten + nine);

int i=1;

System.out.println(i + ten);
```

19 followed by 20

19 followed by 11

Error: Can't convert java lang Integer

d) 10 followed by 1

Ans : c.

INPUT / OUTPUT : EXPLORING JAVA.IO

What is meant by Stream and what are the types of Streams and classes of the Streams?

Ans : A Stream is an abstraction that either produces or consumes information.

There are two types of Streams. They are:

Byte Streams : Byte Streams provide a convenient means for handling input and output of bytes.

Character Streams : Character Streams provide a convenient means for handling input and output of characters.

Byte Stream classes : Byte Streams are defined by using two abstract classes. They are:InputStream and OutputStream.

Character Stream classes : Character Streams are defined by using two abstract classes. They are : Reader and Writer.

Which of the following statements are true?

UTF characters are all 8-bits.

UTF characters are all 16-bits.

UTF characters are all 24-bits.

Unicode characters are all 16-bits.

Bytecode characters are all 16-bits.

Ans : d.

Which of the following statements are true?

When you construct an instance of File, if you do not use the filenaming semantics of the local machine, the constructor will throw an IOException.

When you construct an instance of File, if the corresponding file does not exist on the local file system, one will be created.

When an instance of File is garbage collected, the corresponding file on the local file system is deleted.

None of the above.

Ans : a,b and c.

The File class contains a method that changes the current working directory.

True

False

Ans : b.

It is possible to use the File class to list the contents of the current working directory.

True

False

Ans : a.

Readers have methods that can read and return floats and doubles.

True

False

Ans : b.

You execute the code below in an empty directory. What is the result?

```
File f1 = new File("dirname");
File f2 = new File(f1, "filename");
```

A new directory called dirname is created in the current working directory.

A new directory called dirname is created in the current working directory. A new file called filename is created in directory dirname.

A new directory called dirname and a new file called filename are created, both in the current working directory.

A new file called filename is created in the current working directory.

No directory is created, and no file is created.
Ans : e.

What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?

Ans : The Reader/Writer class hierarchy is character-oriented and the InputStream/OutputStream class hierarchy is byte-oriented.

What is an I/O filter?

Ans : 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.

What is the purpose of the File class?

Ans : The File class is used to create objects that provide access to the files and directories of a local file system.

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

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

What is the difference between the File and RandomAccessFile classes?

Ans : 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.

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

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

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

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

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

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

How many bits are used to represent Unicode, ASCII, UTF-16 and UTF-8 characters?

Ans : Unicode requires 16-bits and ASCII requires 8-bits. Although the ASCII character set uses only 1-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.

Which of the following are true?

The InputStream and OutputStream classes are byte-oriented.
The ObjectInputStream and ObjectOutputStream do not support serialized object input and output.
The Reader and Writer classes are character-oriented.
The Reader and Writer classes are the preferred solution to serialized object output.
Ans : a and c.

Which of the following are true about I/O filters?

Filters are supported on input, but not on output.
Filters are supported by the InputStream/OutputStream class hierarchy, but not by the Reader/Writer class hierarchy.
Filters read from one stream and write to another.
A filter may alter data that is read from one stream and written to another.
Ans : c and d.

Which of the following are true?

Any Unicode character is represented using 16-bits.
7-bits are needed to represent any ASCII character.
UTF-8 characters are represented using only 8-bits.
UTF-16 characters are represented using only 16-bits.
Ans : a and b.

Which of the following are true?

The Serializable interface is used to identify objects that may be written to an output stream.

The Externalizable interface is implemented by classes that control the way in which their objects are serialized.

The Serializable interface extends the Externalizable interface.

The Externalizable interface extends the Serializable interface.

Ans : a, b and d.

Which of the following are true about the File class?

A File object can be used to change the current working directory.

A File object can be used to access the files in the current directory.

When a File object is created, a corresponding directory or file is created in the local file system.

File objects are used to access files and directories on the local file system.

File objects can be garbage collected.

When a File object is garbage collected, the corresponding file or directory is deleted.

Ans : b, d and e.

How do you create a Reader object from an InputStream object?

Use the static createReader( ) method of InputStream class.

Use the static createReader( ) method of Reader class.

Create an InputStreamReader object, passing the InputStream object as an argument to the InputStreamReader constructor.

Create an OutputStreamReader object, passing the InputStream object as an argument to the OutputStreamReader constructor.

Ans : c.

Which of the following are true?

Writer classes can be used to write characters to output streams using different character encodings.

Writer classes can be used to write Unicode characters to output streams.

Writer classes have methods that support the writing of the values of any Java primitive type to output streams.

Writer classes have methods that support the writing of objects to output streams.

Ans : a and b.

The isFile( ) method returns a boolean value depending on whether the file object is a file or a directory.

True.

False.

Ans : a.

Reading or writing can be done even after closing the input/output source.

True.

False.

Ans : b.

The ________ method helps in clearing the buffer.

Ans : flush( ).

The System.err method is used to print error message.

True.

False.

Ans : a.

What is meant by StreamTokenizer?

Ans : StreamTokenizer breaks up InputStream into tokens that are delimited by sets of characters.

It has the constructor : StreamTokenizer(Reader inStream).

Here inStream must be some form of Reader.

What is Serialization and deserialization?

Ans : Serialization is the process of writing the state of an object to a byte stream.

Deserialization is the process of restoring these objects.

30) Which of the following can you perform using the File class?

a) Change the current directory

b) Return the name of the parent directory

c) Delete a file

d) Find if a file contains text or binary information

Ans : b and c.

31)How can you change the current working directory using an instance of the File class called FileName?

FileName.chdir("DirName").

FileName.cd("DirName").

FileName.cwd("DirName").

The File class does not support directly changing the current directory.

Ans : d.

APPLETS

What is an Applet? Should applets have constructors?

Ans : Applet is a dynamic and interactive program that runs inside a Web page displayed by a Java capable browser. We don’t have the concept of Constructors in Applets.

How do we read number information from my applet’s parameters, given that Applet’s getParameter() method returns a string?

Ans : Use the parseInt() method in the Integer Class, the Float(String) constructor in the Class Float, or the Double(String) constructor in the class Double.

How can I arrange for different applets on a web page to communicate with each other?

Ans : Name your applets inside the Applet tag and invoke AppletContext’s getApplet() method in your applet code to obtain references to the other applets on the page.

How do I select a URL from my Applet and send the browser to that page?

Ans : Ask the applet for its applet context and invoke showDocument() on that context object.

```
Eg. URL targetURL;
String URLString
AppletContext context = getAppletContext();
try{
targetUR L = new URL(URLString);
} catch (Malformed URLException e){
// Code for recover from the exception
}
context. showDocument (targetURL);
```

Can applets on different pages communicate with each other?

Ans : No. Not Directly. The applets will exchange the information at one meeting place either on the local file system or at remote system.

How do Applets differ from Applications?

Ans : Appln: Stand Alone

Applet: Needs no explicit installation on local m/c.

Appln: Execution starts with main() method.

Applet: Execution starts with init() method.

Appln: May or may not be a GUI

Applet: Must run within a GUI (Using AWT)

How do I determine the width and height of my application?

Ans : Use the getSize() method, which the Applet class inherits from the Component class in the Java.awt package. The getSize() method returns the size of the applet as a Dimension object, from which you extract separate width, height fields.

Eg. Dimension dim = getSize ();
int appletwidth = dim.width ();

8) What is AppletStub Interface?

Ans : The applet stub interface provides the means by which an applet and the browser communicate. Your code will not typically implement this interface.

It is essential to have both the .java file and the .html file of an applet in the same directory.

True.
False.
Ans : b.

The <PARAM> tag contains two attributes namely _________ and _______.

Ans : Name , value.

Passing values to parameters is done in the _________ file of an applet.

Ans : .html.

12) What tags are mandatory when creating HTML to display an applet

name, height, width
code, name
codebase, height, width
d) code, height, width
Ans : d.

Applet's getParameter( ) method can be used to get parameter values.

True.
False.
Ans : a.

What are the Applet's Life Cycle methods? Explain them?

Ans : init( ) method - Can be called when an applet is first loaded.
start( ) method - Can be called each time an applet is started.
paint( ) method - Can be called when the applet is minimized or refreshed.
stop( ) method - Can be called when the browser moves off the applet's page.
destroy( ) method - Can be called when the browser is finished with the applet.

What are the Applet's information methods?

Ans : getAppletInfo( ) method : Returns a string describing the applet, its author ,copy
right information, etc.
getParameterInfo( ) method : Returns an array of string describing the applet's parameters.

All Applets are subclasses of Applet.

True.
False.
Ans : a.

All Applets must import java.applet and java.awt.

True.
False.
Ans : a.

What are the steps involved in Applet development?

Ans : a) Edit a Java source file,
b) Compile your program and
c) Execute the appletviewer, specifying the name of your applet's source file.

Applets are executed by the console based Java run-time interpreter.

True.
False.
Ans : b.

Which classes and interfaces does Applet class consist?

Ans : Applet class consists of a single class, the Applet class and three interfaces: AppletContext,
AppletStub and AudioClip.

What is the sequence for calling the methods by AWT for applets?

Ans : When an applet begins, the AWT calls the following methods, in this sequence.
init( )
start( )
paint( )
When an applet is terminated, the following sequence of method cals takes place :
stop( )
destroy( )

Which method is used to output a string to an applet?

Ans : drawString ( ) method.

Every color is created from an RGB value.

True.
False
Ans : a.

EVENT HANDLING

The event delegation model, introduced in release 1.1 of the JDK, is fully compatible with the
event model.

True
False
Ans : b.

A component subclass that has executed enableEvents( ) to enable processing of a certain kind of event cannot also use an adapter as a listener for the same kind of event.

True
False
Ans : b.

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

Ans : The java.util.eventObject class is the highest-level class in the event-delegation hierarchy.

What interface is extended by AWT event listeners?

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

What class is the top of the AWT event hierarchy?

Ans : The java.awt.AWTEvent class is the highest-level class in the AWT event class hierarchy.

What event results from the clicking of a button?

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

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

Ans : 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.

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

Ans : 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.

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

Ans : The event-delegation has two advantages over the event-inheritance model. They are :

It enables event handling by objects other than the ones that generate the events. This
allows a clean separation between a component's design and its use.

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.

What is the purpose of the enableEvents( ) method?

Ans :The enableEvents( ) method is used to enable an event for a particular object.

Which of the following are true?

The event-inheritance model has replaced the event-delegation model.

The event-inheritance model is more efficient than the event-delegation model.

The event-delegation model uses event listeners to define the methods of event-handling classes.

The event-delegation model uses the handleEvent( ) method to support event handling.

Ans : c.

Which of the following is the highest class in the event-delegation model?

java.util.EventListener

java.util.EventObject

java.awt.AWTEvent

java.awt.event.AWTEvent

Ans : b.

When two or more objects are added as listeners for the same event, which listener is first invoked to handle the event?

The first object that was added as listener.

The last object that was added as listener.

There is no way to determine which listener will be invoked first.

It is impossible to have more than one listener for a given event.

Ans : c.

Which of the following components generate action events?

Buttons

Labels

Check boxes

Windows

Ans : a.

Which of the following are true?

A TextField object may generate an ActionEvent.

A TextArea object may generate an ActionEvent.

A Button object may generate an ActionEvent.

A MenuItem object may generate an ActionEvent.

Ans : a,c and d.

Which of the following are true?

The MouseListener interface defines methods for handling mouse clicks.

The MouseMotionListener interface defines methods for handling mouse clicks.

The MouseClickListener interface defines methods for handling mouse clicks.

The ActionListener interface defines methods for handling the clicking of a button.

Ans : a and d.

Suppose that you want to have an object eh handle the TextEvent of a TextArea object t. How should you add eh as the event handler for t?

t.addTextListener(eh);
eh.addTextListener(t);
addTextListener(eh.t);
addTextListener(t,eh);
Ans : a.

What is the preferred way to handle an object’s events in Java 2?

Override the object’s handleEvent( ) method.
Add one or more event listeners to handle the events.
Have the object override its processEvent( ) methods.
Have the object override its dispatchEvent( ) methods.
Ans : b.

Which of the following are true?

A component may handle its own events by adding itself as an event listener.
A component may handle its own events by overriding its event-dispatching method.
A component may not handle oits own events.
A component may handle its own events only if it implements the handleEvent( ) method.
Ans : a and b.

How many types of events are provided by AWT? Explain them?

Ans : The AWT provides two types of events. They are :

Low-level event : A low-level event is the one that represents a low-level input or window-system occurrence on a visual component on the screen.

Semantic event : Semantic event is defined at a higher-level to encapsulate the semantics of a user interface component’s model.

A __________ is an object that originates or "fire" events.

Ans : source.

The event listener corresponding to handling keyboard events is the _________ .

Ans : KeyListener.

What are the types of mouse event listeners?

Ans : MouseListener and MouseMotionListener.

24) Which of the following are correct event handling methods

a) mousePressed(MouseEvent e){}
b) MousePressed(MouseClick e){}
c) functionKey(KeyPress k){}
d) componentAdded(ContainerEvent e){}
Ans : a and d.

25) Which of the following are true?

a) A component may have only one event listener attached at a time
b) An event listener may be removed from a component
c) The ActionListener interface has no corresponding Adapter class
d) The processing of an event listener requires a try/catch block
Ans : b and c.

AWT : WINDOWS, GRAPHICS AND FONTS

How would you set the color of a graphics context called g to cyan?

g.setColor(Color.cyan);
g.setCurrentColor(cyan);
g.setColor("Color.cyan");
g.setColor("cyan’);
g.setColor(new Color(cyan));
Ans : a.

The code below draws a line. What color is the line?

```
g.setColor(Color.red.green.yellow.red.cyan);
g.drawLine(0, 0, 100,100);
```

Red
Green
Yellow
Cyan
Black
Ans : d.

What does the following code draw?

```
g.setColor(Color.black);
g.drawLine(10, 10, 10, 50);
g.setColor(Color.RED);
g.drawRect(100, 100, 150, 150);
```

A red vertical line that is 40 pixels long and a red square with sides of 150 pixels
A black vertical line that is 40 pixels long and a red square with sides of 150 pixels
A black vertical line that is 50 pixels long and a red square with sides of 150 pixels
A red vertical line that is 50 pixels long and a red square with sides of 150 pixels
A black vertical line that is 40 pixels long and a red square with sides of 100 pixel
Ans : b.

Which of the statements below are true?
A polyline is always filled.
b) A polyline can not be filled.
c) A polygon is always filled.
d) A polygon is always closed
e) A polygon may be filled or not filled
Ans : b, d and e.

What code would you use to construct a 24-point bold serif font?
new Font(Font.SERIF, 24,Font.BOLD);
new Font("SERIF", 24, BOLD");
new Font("BOLD ", 24,Font.SERIF);
new Font("SERIF", Font.BOLD,24);
new Font(Font.SERIF, "BOLD", 24);
Ans : d.

What does the following paint( ) method draw?

```
Public void paint(Graphics g) {
g.drawString("question #6",10,0);
}
```

The string "question #6", with its top-left corner at 10,0
A little squiggle coming down from the top of the component, a little way in from the left edge
Ans : b.

What does the following paint( ) method draw?

```
Public void paint(Graphics g) {
g.drawString("question #6",10,0);
}
```

A circle at (100, 100) with radius of 44
A circle at (100, 44) with radius of 100
A circle at (100, 44) with radius of 44
The code does not compile
Ans : d.

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

What are the Component subclasses that support painting.
Ans : The Canvas, Frame, Panel and Applet classes support painting.
What is the difference between the paint( ) and repaint( ) method?
Ans : The paint( ) method supports painting via a Graphics object. The repaint( ) method is used
to cause paint( ) to be invoked by the AWT painting method.
What is the difference between the Font and FontMetrics classes?
Ans : The FontMetrics class is used to define implementation-specific properties, such as ascent
and descent, of a Font object.
Which of the following are passed as an argument to the paint( ) method?
A Canvas object
A Graphics object
An Image object
A paint object
Ans : b.
Which of the following methods are invoked by the AWT to support paint and repaint operations?
paint( )
repaint( )
draw( )
redraw( )
Ans : a.
Which of the following classes have a paint( ) method?
Canvas
Image
Frame
Graphics
Ans : a and c.
Which of the following are methods of the Graphics class?
drawRect( )
drawImage( )
drawPoint( )
drawString( )
Ans : a, b and d.
Which Font attributes are available through the FontMetrics class?
ascent
leading
case
height
Ans : a, b and d.
Which of the following are true?
The AWT automatically causes a window to be repainted when a portion of a window has been minimized and then maximized.
The AWT automatically causes a window to be repainted when a portion of a window has been covered and then uncovered.
The AWT automatically causes a window to be repainted when application data is changed.
The AWT does not support repainting operations.
Ans : a and b.
Which method is used to size a graphics object to fit the current size of the window?
Ans : getSize( ) method.
What are the methods to be used to set foreground and background colors?
Ans : setForeground( ) and setBackground( ) methods.
19) You have created a simple Frame and overridden the paint method as follows

```
public void paint(Graphics g){

 g.drawString("Dolly",50,10);

}
```

What will be the result when you attempt to compile and run the program?

The string "Dolly" will be displayed at the centre of the frame

b) An error at compilation complaining at the signature of the paint method

c) The lower part of the word Dolly will be seen at the top of the form, with the top hidden.

d) The string "Dolly" will be shown at the bottom of the form

Ans : c.

20) Where g is a graphics instance what will the following code draw on the screen.

```
g.fillArc(45,90,50,50,90,180);
```

a) An arc bounded by a box of height 45, width 90 with a centre point of 50,50, starting at an angle of 90 degrees traversing through 180 degrees counter clockwise.

b) An arc bounded by a box of height 50, width 50, with a centre point of 45,90 starting at an angle of 90 degrees traversing through 180 degrees clockwise.

c) An arc bounded by a box of height 50, width 50, with a top left at coordinates of 45, 90, starting at 90 degrees and traversing through 180 degrees counter clockwise.

d) An arc starting at 45 degrees, traversing through 90 degrees clockwise bounded by a box of height 50, width 50 with a centre point of 90, 180.

Ans : c.

21) Given the following code

```
import java.awt.*;
public class SetF extends Frame{
public static void main(String argv[]){
SetF s = new SetF();
s.setSize(300,200);
s.setVisible(true);
}
}
```

How could you set the frame surface color to pink

a)s.setBackground(Color.pink);

b)s.setColor(PINK);

c)s.Background(pink);

d)s.color=Color.pink

Ans : a.

AWT: CONTROLS, LAYOUT MANAGERS AND MENUS

What is meant by Controls and what are different types of controls?

Ans : Controls are componenets that allow a user to interact with your application.

The AWT supports the following types of controls:

Labels
Push buttons
Check boxes
Choice lists
Lists

Scroll bars
Text components
These controls are subclasses of Component.
You want to construct a text area that is 80 character-widths wide and 10 character-heights tall. What code do you use?
new TextArea(80, 10)
new TextArea(10, 80)
Ans: b.
A text field has a variable-width font. It is constructed by calling new TextField("iiiii"). What happens if you change the contents of the text field to "wwwww"? (Bear in mind that is one of the narrowest characters, and w is one of the widest.)
The text field becomes wider.
The text field becomes narrower.
The text field stays the same width; to see the entire contents you will have to scroll by using the and keys.
The text field stays the same width; to see the entire contents you will have to scroll by using the text field’s horizontal scroll bar.
Ans : c.
The CheckboxGroup class is a subclass of the Component class.
True
False
Ans : b.
5) What are the immediate super classes of the following classes?
a) Container class
b) MenuComponent class
c) Dialog class
d) Applet class
e) Menu class
Ans : a) Container - Component
b) MenuComponent - Object
c) Dialog - Window
d) Applet - Panel
e) Menu - MenuItem
6) What are the SubClass of Textcomponent Class?
Ans : TextField and TextArea
7) Which method of the component class is used to set the position and the size of a component?
Ans : setBounds()
8) Which TextComponent method is used to set a TextComponent to the read-only state?
Ans : setEditable()
9) How can the Checkbox class be used to create a radio button?
Ans : By associating Checkbox objects with a CheckboxGroup.
10) What Checkbox method allows you to tell if a Checkbox is checked?
Ans : getState()
11) Which Component method is used to access a component's immediate Container?
getVisible()
getImmediate
getParent()
getContainer
Ans : c.
12) What methods are used to get and set the text label displayed by a Button object?

Ans : getLabel( ) and setLabel( )
13) What is the difference between a Choice and a List?
Ans : 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.
14) Which Container method is used to cause a container to be laid out and redisplayed?
Ans : validate( )
15) What is the difference between a Scollbar and a Scrollpane?
Ans : A Scrollbar is a Component, but not a Container.
A Scrollpane is a Container and handles its own events and performs its own scrolling.
16) Which Component subclass is used for drawing and painting?
Ans : Canvas.
17) Which of the following are direct or indirect subclasses of Component?
Button
Label
CheckboxMenuItem
Toolbar
Frame
Ans : a, b and e.
18) Which of the following are direct or indirect subclasses of Container?
Frame
TextArea
MenuBar
FileDialog
Applet
Ans : a,d and e.
19) Which method is used to set the text of a Label object?
setText( )
setLabel( )
setTextLabel( )
setLabelText( )
Ans : a.
20) Which constructor creates a TextArea with 10 rows and 20 columns?
new TextArea(10, 20)
new TextArea(20, 10)
new TextArea(new Rows(10), new columns(20))
new TextArea(200)
Ans : a.
(Usage is TextArea(rows, columns)
21) Which of the following creates a List with 5 visible items and multiple selection enabled?
new List(5, true)
new List(true, 5)
new List(5, false)
new List(false,5)
Ans : a.
[Usage is List(rows, multipleMode)]
22) Which are true about the Container class?
The validate( ) method is used to cause a Container to be laid out and redisplayed.
The add( ) method is used to add a Component to a Container.
The getBorder( ) method returns information about a Container’s insets.

The getComponent( ) method is used to access a Component that is contained in a Container.
Ans : a, b and d.
23) Suppose a Panel is added to a Frame and a Button is added to the Panel. If the Frame’s font is set to 12-point TimesRoman, the Panel’s font is set to 10-point TimesRoman, and the Button’s font is not set, what font will be used to dispaly the Button’s label?
12-point TimesRoman
11-point TimesRoman
10-point TimesRoman
9-point TimesRoman
Ans : c.
A Frame’s background color is set to Color.Yellow, and a Button’s background color is to Color.Blue. Suppose the Button is added to a Panel, which is added to the Frame. What background color will be used with the Panel?
Colr.Yellow
Color.Blue
Color.Green
Color.White
Ans : a.
25) Which method will cause a Frame to be displayed?
show( )
setVisible( )
display( )
displayFrame( )
Ans : a and b.
26) All the componenet classes and container classes are derived from _________ class.
Ans : Object.
27) Which method of the container class can be used to add components to a Panel.
Ans : add ( ) method.
28) What are the subclasses of the Container class?
Ans : The Container class has three major subclasses. They are :
Window
Panel
ScrollPane
29) The Choice component allows multiple selection.
True.
False.
Ans : b.
30) The List component does not generate any events.
True.
False.
Ans : b.
31) Which components are used to get text input from the user.
Ans : TextField and TextArea.
32) Which object is needed to group Checkboxes to make them exclusive?
Ans : CheckboxGroup.
33) Which of the following components allow multiple selections?
Non-exclusive Checkboxes.
Radio buttons.
Choice.
List.
Ans : a and d.
34) What are the types of Checkboxes and what is the difference between them?

Ans : Java supports two types of Checkboxes. They are : Exclusive and Non-exclusive. In case of exclusive Checkboxes, only one among a group of items can be selected at a time. I f an item from the group is selected, the checkbox currently checked is deselected and the new selection is highlighted. The exclusive Checkboxes are also called as Radio buttons.

The non-exclusive checkboxes are not grouped together and each one can be selected independent of the other.

35) What is a Layout Manager and what are the different Layout Managers available in java.awt and what is the default Layout manager for the panal and the panal subclasses?

Ans: A layout Manager is an object that is used to organize components in a container.

The different layouts available in java.awt are :

FlowLayout, BorderLayout, CardLayout, GridLayout and GridBag Layout.

The default Layout Manager of Panal and Panal sub classes is FlowLayout".

36) Can I exert control over the size and placement of components in my interface?

Ans : Yes.

```
myPanal.setLayout(null);
myPanal.setbounds(20,20,200,200);
```

37) Can I add the same component to more than one container?

Ans : No. Adding a component to a container automatically removes it from any previous parent(container).

38) How do I specify where a window is to be placed?

Ans : Use setBounds, setSize, or setLocation methods to implement this.

```
setBounds(int x, int y, int width, int height)
setBounds(Rectangle r)
setSize(int width, int height)
setSize(Dimension d)
setLocation(int x, int y)
setLocation(Point p)
```

39) How can we create a borderless window?

Ans : Create an instance of the Window class, give it a size, and show it on the screen.

```
eg. Frame aFrame = ......
Window aWindow = new Window(aFrame);
aWindow.setLayout(new FlowLayout());
aWindow.add(new Button("Press Me"));
aWindow.getBounds(50,50,200,200);
aWindow.show();
```

40) Can I create a non-resizable windows? If so, how?

Ans: Yes. By using setResizable() method in class Frame.

41) What is the default Layout Manager for the Window and Window subclasses (Frame,Dialog)?

Ans : BorderLayout().

42) How are the elements of different layouts organized?

Ans : FlowLayout : The elements of a FlowLayout are organized in a top to bottom, left to right fashion.

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

CardLayout : The elements of a CardLayout are stacked, one on top of the other, like a deck of cards.

GridLayout : The elements of a GridLayout are of equal size and are laid out using the square of a grid._
GridBagLayout : 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._

43) Which containers use a BorderLayout as their default layout?
Ans : The Window, Frame and Dialog classes use a BorderLayout as their default layout.

44) Which containers use a FlowLayout as their default layout?
Ans : The Panel and the Applet classes use the FlowLayout as their default layout.

45) What is the preferred size of a component?
Ans : The preferred size of a component size that will allow the component to display normally.

46) Which method is method to set the layout of a container?
startLayout( )
initLayout( )
layoutContainer( )
setLayout( )
Ans : d.

47) Which method returns the preferred size of a component?
getPreferredSize( )
getPreferred( )
getRequiredSize( )
getLayout( )
Ans : a.

48) Which layout should you use to organize the components of a container in a tabular form?
CardLayout
BorederLayout
FlowLayout
GridLayout
Ans : d.

An application has a frame that uses a Border layout manager. Why is it probably not a good idea to put a vertical scroll bar at North in the frame?
The scroll bar’s height would be its preferred height, which is not likely to be enough.
The scroll bar’s width would be the entire width of the frame, which would be much wider than necessary.
Both a and b.
Neither a nor b. There is no problem with the layout as described.
Ans : c.

What is the default layouts for a applet, a frame and a panel?
Ans : For an applet and a panel, Flow layout is the default layout, whereas Border layout is default layout for a frame.

If a frame uses a Grid layout manager and does not contain any panels, then all the components within the frame are the same width and height.
True
False.
Ans : a.

If a frame uses its default layout manager and does not contain any panels, then all the components within the frame are the same width and height.
True
False.
Ans : b.
With a Border layout manager, the component at Center gets all the space that is left over, after the components at North and South have been considered.
True
False
Ans : b.
An Applet has its Layout Manager set to the default of FlowLayout. What code would be the correct to change to another Layout Manager?
setLayoutManager(new GridLayout());
setLayout(new GridLayout(2,2));
c) setGridLayout(2,2,))
d setBorderLayout();
Ans : b.
55) How do you indicate where a component will be positioned using Flowlayout?
a) North, South,East,West
b) Assign a row/column grid reference
c) Pass a X/Y percentage parameter to the add method
d) Do nothing, the FlowLayout will position the component
Ans :d.

56) How do you change the current layout manager for a container?
a) Use the setLayout method
b) Once created you cannot change the current layout manager of a component
c) Use the setLayoutManager method
d) Use the updateLayout method
Ans :a.
57)When using the GridBagLayout manager, each new component requires a new instance of the GridBagConstraints class. Is this statement true or false?
a) true
b) false
Ans : b.
58) Which of the following statements are true?
a)The default layout manager for an Applet is FlowLayout
b) The default layout manager for an application is FlowLayout
c) A layout manager must be assigned to an Applet before the setSize method is called
d) The FlowLayout manager attempts to honor the preferred size of any components
Ans : a and d.
59) Which method does display the messages whenever there is an item selection or deselection of the CheckboxMenuItem menu?
Ans : itemStateChanged method.
60) Which is a dual state menu item?
Ans : CheckboxMenuItem.
61) Which method can be used to enable/diable a checkbox menu item?
Ans : setState(boolean).
Which of the following may a menu contain?
A separator
A check box
A menu

A button
A panel
Ans : a and c.
Which of the following may contain a menu bar?
A panel
A frame
An applet
A menu bar
A menu
Ans : b
64) What is the difference between a MenuItem and a CheckboxMenuItem?
Ans : The CheckboxMenuItem class extends the MenuItem class to support a menu item
that may be checked or unchecked.

65) Which of the following are true?
A Dialog can have a MenuBar.
MenuItem extends Menu.
A MenuItem can be added to a Menu.
A Menu can be added to a Menu.
Ans : c and d.
Q1.1
What is an object?
See also: Q1.2, Q1.3, Q1.7

Q1.2
What is a class?
See also: Q1.1, Q1.3, Q1.7, Q1.12, Q1.20, Q1.32, Q2.19, JLS p. 38

Q1.3
What is a method?
See also: Q1.2, Q1.4, Q1.6, Q1.14, Q2.10

Q1.4
What is the signature of a method?
See also: Q1.13, Q1.14, Q2.24

Q1.5
What is the difference between an instance variable and a class variable?
See also: Q1.6

Q1.6
What is the difference between an instance method and a class method?
VariableExample.html
See also: Q1.2, Q1.3, Q1.5

Q1.7
How do I create an instance of a class?
See also: Q1.10, Q2.15, Q3.5, Q3.7, Q10.4, Q10.16

Q1.8
What is an abstract method?
See also: Q1.9, Q1.11

Q1.9

What is an abstract class?
See also: Q1.8, Q1.11, Q5.1

Q1.10
What is an object reference?
See also: Q2.8, JLS p. 39

Subclassing, Overloading, and Overriding

Q1.11
What is a subclass?
See also: Q1.12, Q1.25

Q1.12
What is inheritance?
See also: Q1.5, Q1.6, Q5.1

Q1.13
What is an overloaded method?
See also: Q1.2, Q1.3, Q1.4, Q1.12, Q1.14

Q1.14
What does it mean to override a method?
OverrideMethodExample.html
See also: Q1.4, Q1.12, Q1.16, Q1.17, Q2.26, Q5.1, Q8.5

Q1.15
What is the difference between overloading and overriding?
See also: Q1.13, Q1.14, Q1.16

Q1.16
Can I override the equals method or clone method from class Object to take a parameter or return a value of the type that I specify?
See also: Q1.4, Q1.13, Q1.14, Q1.15, Q2.9

Q1.17
What is the super keyword used for?
SuperExample.html
See also: Q1.11, Q1.12

Q1.18
Does the Java language provide virtual methods, like C++?
See also: Q1.10, Q1.12, Q1.14, Q1.19, Q1.26

Q1.19
What is a final class?
See also: Q1.18

Interfaces and Abstract Classes

Q1.20

What is an interface?

See also: Q1.2, Q1.21, Q1.24, JLS pp. 186 188

Q1.21

How does a class implement an interface?

See also: Q1.9, Q1.20, Q1.22, Q9.16

Q1.22

Can I instantiate an interface?

InstantiateInterfaceExample.html

See also: Q1.7, Q1.20, Q1.21

Q1.23

Why does a method in an interface appear to be public even though I didn't declare it to be public?

See also: Q1.8, Q1.20, Q1.32, JLS p. 187

Q1.24

How is an abstract class different from an interface?

See also: Q1.9, Q1.20, Q1.25

Q1.25

Does the Java language allow multiple inheritance?

See also: Q1.2, Q1.11, Q1.12, Q1.20

Q1.26

What is the instanceof keyword, and what does it do?

InstanceofExample.html

Q1.27

Why do I get the error message Can't access protected method clone... when I try to clone an object?

See also: Q1.28, Q1.33

Q1.28

How do I design a class so that it supports cloning?

CloneExample.html

See also: Q1.27, Q1.32, Q1.33

Packages and Access Modifiers

Q1.29

What are packages, and what are they used for?

See also: Q1.2, Q1.20, Q1.30, Q1.32, JLS p. 119

Q1.30

I've seen both java.applet.Applet and Applet used to refer to the Applet class -- what's the difference?

See also: Q1.29, Q1.31, JLS pp. 120ff

Q1.31

Why can I get some simple class names "for free," without using an import declaration?

See also: Q1.29, Q1.30, JLS p. 119

Q1.32

Is there a default access modifier for classes and interfaces? For class members (methods, constructors, and fields)?

See also: Q1.27, Q1.29, Q1.33

Q1.33

What does protected access mean?

See also: Q1.29, Q1.32, JLS p. 100

Q1.34

What is the accessibility of a public method or field inside a nonpublic class or interface?

See also: Q1.18, Q1.29, Q1.32

Outline

Constants and Expressions (Q2.1--Q2.10)

Variables and Methods (Q2.11--Q2.18)

Arrays (Q2.19--Q2.21)

Exceptions (Q2.22--Q2.26)

Constants and Expressions

Q2.1

What is the difference between Integer and int in Java -- why do I get the following error: Can't convert java.lang.Integer to int?

See also: Q2.11

Q2.2

How do I treat all 8 bits of a byte as an unsigned quantity?

UnsignedByteExample.html

See also: Q2.1, JLS pp. 61 75

Q2.3

How do I work around Java's lack of true enums?

See also: Q2.1

Q2.4

Why is goto a reserved keyword in Java -- shouldn't it be outlawed?

KeywordExample.html

See also: JLS p. 18, JLS pp. 283--286

Q2.5

What are some guidelines for using uppercase and lowercase letters in my identifiers?

See also: JLS pp. 106 111

Q2.6

Is there any limit to the length of an identifier?

See also: JVMS p. 100

Q2.7
Why doesn't Java have user-defined operator overloading?
See also: Q11.1

Q2.8
Why do I have to put an f after a floating point constant?
See also: Q2.9

Q2.9
How and when can I cast from one class to another?
See also: Q1.10, Q1.26, Q2.8, Q2.11, Q2.15

Q2.10
Can I use C-like data structures in Java?
See also: Q1.1, Q1.2, Q1.3

Variables and Methods

Q2.11
Are Java objects pointers?
See also: Q1.1, Q1.2, Q1.10, Q2.1, Q2.12, Q2.13

Q2.12
In a method invocation, does Java pass arguments by reference or by value?
See also: Q1.10, Q2.1, Q2.11, Q11.1

Q2.13
If the Java language lacks pointers, how do I implement classic pointer structures like linked lists?
LinkedListExample.html
See also: Q1.10, Q2.11, Q2.12

Q2.14
What does the following error message mean: Can't make a static reference to nonstatic variable?
See also: Q1.5, Q1.6

Q2.15
In a class method, how can I get the name of the class, or create a new instance of the class?
ClassClassExample.html
See also: Q1.30, Q2.16, Q3.7

Q2.16
Can I write a method that delivers dynamically (at run time) all public methods of an object?
GetMethod11Example.html
See also: Q2.17, Q2.18

Q2.17
Can I invoke methods dynamically, from names (String instances) that are determined at run time?
InvokeMethodDynamically11Example.html
See also: Q2.1, Q2.16, Q2.22, Q2.23

Q2.18
How can I accomplish the equivalent of function pointers in Java, for instance, for use in an array?
MethodTable11Example.html
See also: Q7.2

Arrays

Q2.19
Can I allocate an array dynamically?
See also: Q2.20, JLS p. 193

Q2.20
How do I initialize an array of objects?
InitializeObjectArrayExample.html
See also: Q2.11

Q2.21
If arrays are objects, why can't I use a length method to determine an array's size?

Exceptions

Q2.22
What is an exception?
ExceptionExample.html
See also: Q2.23

Q2.23
Why does the compiler complain about InterruptedException when I try to use Thread's sleep method?
See also: Q2.22, Q2.24

Q2.24
Why do methods have to declare the exceptions they can throw?
See also: Q2.22, Q2.23, Q2.25

Q2.25
What's the difference between a runtime exception and a plain exception -- why don't runtime exceptions have to be declared?
See also: Q2.24, Q2.25
Virtual Machine (Q3.1--Q3.9)

Virtual Machine

Q3.1
When, and by whom, is the main method of a class invoked?
MainExample.html, SimpleTimer.html
See also: Q1.2, Q1.3

Q3.2
What are bytecodes?
See also: Q3.3

Q3.3

What is javap?
See also: Q3.2

Q3.4
What does it mean to say that Java is interpreted?
See also: Q3.2, Q3.3, JVMS Ch. 4

Q3.5
What kind of garbage collection does the Java Virtual Machine use?
See also: Q1.10, Q2.11

Q3.6
Is finalization broken -- why does my finalize method never seem to get invoked?
See also: Q8.20

Q3.7
I'm having trouble invoking methods on the objects returned from Class's forName method -- how should I use Class.forName?
See also: Q2.9

Q3.8
Why do I get verifier errors when loading a class file produced by javac?
See also: Q3.3

Q3.9
How fast are Java programs compared to equivalent C or C++ programs?

Applets versus Applications

Q4.1
What is an applet?
See also: Q4.7, Q4.24, JavaSoft's Security FAQ web page

Q4.2
How do applets differ from applications?
See also: Q4.3, Q4.16, JavaSoft's Security FAQ

Q4.3
Can I write Java code that works both as an applet and as a stand-alone application?
See also: Q3.1, Q4.16, Q4.17, Q4.18, Q4.20, Q11.8, JavaSoft's Security FAQ

Installing Applets

Q4.4
How do I put an applet into a web page?
See also: Q4.7

Q4.5
Can I put more than one applet in a web page?
See also: Q4.20

Q4.6
How do I use the APPLET tag if I want to show users of non-Java-enabled browsers what they're missing?
See also: Q4.4, Q4.7

Q4.7
What is the complete syntax for using the APPLET tag?
See also: the applet TAG page

Q4.8
What are the different pieces that can make up an applet, and how do I install them?
See also: Q4.9, Q4.10

Q4.9
Where can I put my applet's class files, and how do I indicate their location using the APPLET tag?
See also: Q4.8, Q4.10

Q4.10
Can I put my reusable custom classes in a special place so that many different applets can use them?
See also: Q4.8, Q4.9

Applet User Interface

Q4.11
How do I determine the width and height of my applet?
See also: Q5.1

Q4.12
How do I set the background color within the applet area?
AppletSizeExample.html
See also: Q4.13, Q5.1, Q8.9

Q4.13
How can I create a transparent background for my applet?
MatchBackgroundExample.html
See also: Q8.7

Q4.14
Can I put menus and a menu bar on my applet?
AppletMenuExample.html
See also: Q6.8, Q7.8

Q4.15
I know that cursors can be changed from within frames, but how do I change the cursor in my applet?
ChangeCursorExample.html, ChangeCursor11Example.html
See also: Q5.14

Applet Program Structure

Q4.16

Several applet methods seem special, in that I need to define them even if my own code doesn't invoke them -- what are the methods, and when (and by whom) are they invoked?
See also: Q4.17, Q4.18, Q8.1, Q8.5

Q4.17
Should applets have constructors?
See also: Q4.16, Q4.18

Q4.18
How can my applet tell when a user leaves or returns to the web page containing my applet?
InitStartStopExample.html
See also: Q4.16, Q4.17, Q9.2, Q9.3

Q4.19
How do I read number information from my applet's parameters, given that Applet's getParameter method returns a String?
NumericalParametersExample.html
See also: Q10.4

Applet Communication

Q4.20
How can I arrange for different applets on a web page to communicate with each other?
InterappletExample.html
See also: Q4.7

Q4.21
How do I select a URL from my applet and send the browser to that page?
See also: Q10.14

Q4.22
Can applets on different pages communicate with each other?
See also: Q4.20

Miscellaneous

Q4.23
Can I load an applet dynamically into a Java application, and if so, how does the applet get the parameter information it would normally get from the APPLET tag?
See also: Q5.1, Q5.2

Q4.24
Do I need any special server software or setup to deliver applets?
See also: Q4.1

Outline

Components, Containers, and Peers

Q5.1
What are the Component and Container classes?
ButtonCanvasExample.html
See also: Q5.2

Q5.2
How do I control the positioning of my interface components?
See also: Q5.1, Q5.3, Q5.4, Q5.5

Q5.3
What are inset values, and how do I set them?
See also: Q5.1, Q5.2, Q5.8, Q5.12

Q5.4
Can I exert complete control over the size and placement of components in my interface?

Q5.5
What do the invalidate and validate methods do?
See also: Q5.1, Q5.2, Q5.6, Q5.7

Q5.6
Can I add new AWT components to objects already visible on the screen?
CreateButtonsExample.html
See also: Q5.5, Q5.7

Q5.7
Can I add the same component to more than one container?
See also: Q5.1

Q5.8
How can I place an outline around a group of components to show explicitly how the components are grouped?
BorderedPanelExample.html
See also: Q5.2, Q5.3

Q5.9
What are AWT peers?
AwtPeerExample.html
See also: Q5.9

Windows, Frames, and Dialogs

Q5.11
How do I specify where a window is to be placed?
PlaceFrameExample.html, PlaceFrame11Example.html
See also: Q5.2, Q5.12

Q5.12

How can I draw at the top-left corner of a frame without it being covered by the frame's border?

Q5.13

How do I create a borderless window?

See also: Q5.14

Q5.14

Several operations in the AWT, such as setting the cursor (in JDK 1.0.2) or creating a dialog box, require specifying a Frame instance -- how do I determine the Frame instance containing the current component?

GetFrameExample.html

See also: Q4.15

Q5.15

How do I use a FileDialog object in my applet or application?

FileDialogExample.html

Q5.16

Can I create nonresizable windows?

Miscellaneous

Q5.17

What does Component's requestFocus method do?

FocusExample.html

See also: Q6.6, Q7.13

Q5.18

Does the AWT allow you to control the mouse location from within an application?

Q5.19

Does the AWT provide a standard way to signal a user error by flashing, beeping, or some other means?

Beep11Example.html

Q5.20

What fonts are available to my AWT program?

See also: Q5.21

Q5.21

How can I dynamically change font attributes, for instance, rendering a string in successively larger font sizes?

ChangingFontsExample.html

See also: Q5.20

Outline

Events -- JDK 1.0.2 (Q6.1--6.11)

Events -- JDK 1.0.2

Q6.1

What information can be carried in the JDK 1.0.2 by an Event object?

Q6.2

What information do specific event types in the JDK 1.0.2 carry?
See also: Q5.9, Q6.3

Q6.3
What is the general model in the JDK 1.0.2 for distributing and handling events?
See also: Q5.1, Q5.9, Q6.2, Q6.4

Q6.4
What methods should I use in the JDK 1.0.2 to handle events?
See also: Q1.17, Q6.3, Q6.5

Q6.5
What is an action event?
ActionEventExample.html
See also: Q6.3, Q6.4, Q6.6, Q6.8, Q7.7

Q6.6
How does the JDK 1.0.2 handle events for function keys, arrow keys, and so on?
ActionKeyExample.html
See also: Q6.1, Q6.2, Q6.3, Q6.4, Q7.13

Q6.7
My frame doesn't close when I click on Quit/Close in the main menu -- how do I fix this using JDK 1.0.2?
CloseFrameExample.html
See also: Q7.11

Q6.8
How is my program notified when a menu item is selected?
MenuItemEventExample.html
See also: Q5.1, Q6.3, Q6.4, Q6.5, Q7.8

Q6.9
How is my program notified in the JDK 1.0.2 when a list item is selected or deselected?
ListEventExample.html
See also: Q6.2, Q6.4, Q6.5, Q7.9

Q6.10
How do I hook up a scroll bar in the JDK 1.0.2 so that it controls the scrolling of some other component?
ScrollbarExample.html
See also: Q6.2, Q6.4, Q7.10

Q6.11
Does the AWT in the JDK 1.0.2 distinguish between mouse clicks made with different buttons on a two- or three-button mouse?
MultiButtonMouseExample.html
See also: Q7.14

Outline
Event Classes, Listeners, and Methods (Q7.1--Q7.6)
Semantic Events (Q7.7--Q7.10)
Low-Level Events (Q7.11--Q7.14)

Event Classes, Listeners, and Methods

Q7.1
What information do specific event types in the JDK 1.1 carry?
See also: Q7.2, Q7.3

Q7.2
How do I catch events in the JDK 1.1 event model?
EventListener11Example.html
See also: Q1.20, Q7.3

Q7.3
What are the different kinds of event listeners, and what are their methods?
See also: Q6.3, Q7.1, Q7.2, Q7.4

Q7.4
When should I use an event adapter class?
See also: Q7.2, Q7.3

Q7.5
Do events propagate in the JDK 1.1 as they did in the older AWT event model?
PropagateEvent11Example.html
See also: Q6.3, Q7.2, Q7.3

Q7.6
Will my code written for the older JDK 1.0.2 event model still run in the JDK 1.1?
See also: Q6.3, Q6.4

SEMANTIC EVENTS

Q7.7
How do I handle action events in the JDK 1.1?
ActionEvent11Example.html
See also: Q6.5, Q7.2, Q7.3

Q7.8
How do I catch menu item events in the 1.1 AWT event model?
MenuItemEvent11Example.html
See also: Q6.8, Q7.7

Q7.9
In the JDK 1.1 AWT event model, how is my program notified when a list item is selected or deselected?
ListEvent11Example.html
See also: Q6.9, Q7.7

Q7.10
How do I control scrolling in the 1.1 AWT?
ScrollPane11Example.html
See also: Q6.10

LOW-LEVEL EVENTS

Q7.11

My frame doesn't close when I click on Quit/Close in the main menu -- how do I fix this using the JDK 1.1?
See also: Q6.7

Q7.12
Is there an event type that signals when a window is resized?
WindowResizeEvent11Example.html
See also: Q7.11

Q7.13
How do I handle events for function keys, arrow keys, and so on in the JDK 1.1 event model?
ActionKey11Example.html
See also: Q6.6

Q7.14
How does the JDK 1.1 distinguish between mouse clicks made with different buttons on a two- or three-button mouse?
MultiButtonMouse11Example.html
See also: Q6.11

Outline

Drawing AWT Components

Q8.1
What is the paint method for, when is it invoked, and by whom?
PaintExample.html
See also: Q5.1, Q8.2, Q8.3, Q8.5, Q8.6, Q9.17

Q8.2
What should I put in my paint method?
SlowPaintExample.html
See also: Q8.18

Q8.3
What is repaint for, when is it invoked, and by whom?
RepaintExample.html
See also: Q8.1, Q8.4

Q8.4
Why do my repeated calls to repaint not have any effect?
BusyRepaintExample.html
See also: Q9.5, Q9.13, Q9.15

Q8.5
What is update for, when is it invoked, and by whom?
UpdateExample.html
See also: Q8.1, Q8.2, Q8.3, Q8.5, Q8.18

Q8.6

What drawing occurs if my applet or other component reappears after being covered by some other window?
See also: Q8.1, Q8.3, Q8.4, Q8.5

Q8.7
Can I implement an invisible or partly transparent component?
See also: Q4.13, Q5.9

Q8.8
How does the XOR drawing mode work?
XORDrawingExample.html
Loading and Drawing Images

Q8.9
How do I load an image from the net into my applet?
See also: Q4.12, Q8.15

Q8.10
How do I load an image from a file in a stand-alone Java application, rather than in an applet?
AppGetImageExample.html
See also: Q4.12

Q8.11
When is an image actually loaded -- why not immediately?
See also: Q8.12

Q8.12
How can I make sure that my images are completely loaded before I check for their data or parameters?
TrackImageExample.html, TrackErrorImageExample.html
See also: Q8.10, Q8.11, Q8.13

Q8.13
Why does my call to Graphics's drawImage method fail to show the image?
FailedDrawImageExample.html, TrackErrorImageExample.html
See also: Q1.20, Q8.12

Q8.14
Can I force Applet's getImage method to make a new connection for each image rather than reusing a cached version of the image?
FlushImageExample.html
See also: Q8.15

Q8.15
How do I draw text over a background image?
TextOnImageExample.html
See also: Q8.14, Q8.16

Q8.16
How do I load and display a transparent GIF image over a background image?
TransparentImageExample.html
See also: Q8.15

Q8.17

How can I create an image from a buffer of raw image data (red, green, and blue values for each pixel)?

MemoryImageExample.html

See also: Q8.22

Q8.18

What is double buffering -- how can I create and draw to an offscreen image?

OffscreenImageExample.html

See also: Q8.5

Q8.19

How can I get at the raw data of an image, such as the pixel value at a given coordinate?

GrabPixelsExample.html

See also: Q2.23, Q8.17, Q9.16

Images -- JDK 1.0.2

Q8.20

Using the JDK 1.0.2, can I clear or reset a clipping rectangle that either I or the system has created?

SubgraphicsExample.html

See also: Q8.23

Q8.21

Using the JDK 1.0.2, can I copy a subarea of one image into another image?

CopyImageSubareaExample.html

See also: Q8.20, Q8.24

Images -- JDK 1.1

Q8.22

How do I control animation with MemoryImageSource?

See also: Q8.14, Q8.17

Q8.23

Using the JDK 1.1, how do I reset a clipping rectangle?

ResetClip11Example.html

See also: Q8.20

Q8.24

What's the best way in JDK 1.1 to draw just a subarea of an image?

See also: Q8.21

The Java(tm) FAQ -- Threads

Copyright Table of Contents Chapters: 1 2 3 4 5 6 7 8 9 10 11

Outline

Creating and Controlling Threads

Q9.1

What is a thread?
See also: Q9.2, Q9.6, Q9.7, Q9.8

Q9.2
How do I create a thread and start it running?
CreateThreadExample.html
See also: Q1.21, Q9.1, Q9.3

Q9.3
How does Thread's stop method work -- can I restart a stopped thread?
See also: Q9.2, Q9.4, Q9.5, Q9.7, Q9.10

Q9.4
How should I stop a thread so that I can start a new thread later in its place?
See also: Q9.2, Q9.3, Q9.16

Q9.5
How do I specify pause times in my program?
SleepExample.html
See also: Q1.6, Q2.23, Q9.10

Thread Interactions

Q9.6
Why is thread synchronization important for multithreaded programs?
See also: Q9.7, Q9.8

Q9.7
What is a monitor?
See also: Q9.8, Q9.10

Q9.8
How does the synchronized keyword work?
SynchronizationExample.html
See also: Q9.7, Q9.8

Q9.9
What objects do static synchronized methods use for locking?
See also: Q1.30, Q2.23, Q9.7, Q9.8

Q9.10
How do the wait and notifyAll/notify methods enable cooperation between threads?
WaitNotifyExample.html
See also: Q9.11

Q9.11
How do I achieve the effect of condition variables if the Java platform provides me with only wait and notifyAll/notify methods?
ConditionVariableExample.html
See also: Q9.10

Q9.12
How do I make one thread wait for one or more other threads to finish?
JoinExample.html
See also: Q9.5, Q9.10

Q9.13
What do I use the yield method for?
See also: Q9.15

Q9.14
Does the Java Virtual Machine protect me against deadlocks?
DeadlockExample.html
See also: Q9.7, Q9.8

User Threads versus System Threads

Q9.15
Why does my multithreaded program run fine on Windows NT/95 but block on Solaris?
See also: Q9.13

Q9.16
Why do so many applets run a copy of themselves in a separate thread rather than just running as they are?
SystemThreadsExample.html
See also: Q4.2, Q4.16, Q9.4, Q9.17

Q9.17
How can it be that putting an applet thread to sleep in the wrong place can block other applets from running?
BlockAWTThreadExample.html
See also: Q4.16, Q6.4, Q7.3, Q8.1, Q8.5, Q9.16

Q9.18
Can I have a thread wait on an event from the operating system?
The Java(tm) FAQ -- Input, Output, and Network
Copyright Table of Contents Chapters: 1 2 3 4 5 6 7 8 9 10 11

Outline
Basic Input and Output (Q10.1--Q10.6)
URL Connections (Q10.7--Q10.15)
Internet Addresses (Q10.16--Q10.18)
Sockets (Q10.19--Q10.27)

Basic Input and Output

Q10.1
How do I read a line of input at a time?
ReadLineExample.html, ReadLine11Example.html
The Java(tm) FAQ -- Miscellaneous
Copyright Table of Contents Chapters: 1 2 3 4 5 6 7 8 9 10 11

Outline
Classes in java.lang and java.util (Q11.1--Q11.4)
Audio (Q11.5--Q11.10)
Miscellaneous (Q11.11--Q11.14)

Classes in java.lang and java.util

Q11.1
Why doesn't String's replace method have any effect when I apply it to a string?
StringReplaceExample.html

Q11.2
How can I access native operating system calls from my Java program?
See also: Q10.6

Q11.3
Does Java provide standard iterator functions for inspecting a collection of objects?
See also: Q1.20

Q11.4
The Math.random method is too limited for my needs -- how can I generate random numbers more flexibly?
RandomExample.html

Audio

Q11.5
How do I play sounds in an applet?
See also: Q8.11, Q11.6, Q11.7

Q11.6
How do I play several audio clips simultaneously?
See also: Q11.5, Q11.7

Q11.7
Can I tell when an audio clip has finished playing?
See also: Q11.5, Q11.6

Q11.8
How can I play audio in a stand-alone application?
See also: Q4.2

Q11.9
What audio formats does the JDK
Which code segment could execute the stored procedure "countRecs()" located in a database server?
A) PreparedStatement pstmt = connection.prepareStatement("countRecs()");
pstmt.execute();
B) Statement stmt = connection.createStatement();
stmt.executeStoredProcedure("countRecs()");
C) Statement stmt = connection.createStatement();
stmt.execute("COUNTRECS()");
D) StoredProcedureStatement spstmt =
connection.createStoredProcedure("countRecs()");
spstmt.executeQuery();
E) CallableStatement cs = con.prepareCall("{call COUNTRECS}");
cs.executeQuery();
JDK 1.2 includes Reference objects such as WeakReference and PhantomReference.
How would you create your own type of Reference object?
A) Extend Reference.
B) Write methods to interact with the garbage collector.
C) You CANNOT create your own Reference type.

D) Implement Reference.
E) Create a class with a Reference attribute.
Which Java API would you use while writing Java objects that act as servers for distributed CORBA objects written in C++?
A) JNI
B) JDBC
C) Infobus
D) JavaIDL
E) RMI
Which of the following commands starts the Java IDL name service on port 1234?
A) nameserver -InitialPort 1234
B) tnameserv -ORBInitialPort 1234
C) cosnaming -COSInitialPort 1234
D) idlregistry -IDLInitialPort 1234
E) nameservice 1234
What is Java serialization?
A) Distributed persistence.
B) The ability to examine the encapsulated data of a class.
C) Remote method invocation.
D) The ability transform the state of an object into bits and resurrect a copy of the object from those bits.
E) Marshaling and unmarshaling of remote objects.

How would you create a menu item "Save" with a shortcut key of "Ctrl+S"?
A) JMenuItem save = new JMenuItem("Save"); save.addShortcutKey(new KeyStroke(KeyStroke.S | KeyStroke.CTRL_KEY));
B) JMenuItem save = new JMenuItem("Save"); save.setMnemonic("Ctrl+S");
C) You would have to override the KeyPressed event of the top level Frame, and handle the "Ctrl+S" to call the menu item's actionPerformed.
D) JMenuItem save = new JMenuItem("Save"); save.setAccelerator(KeyStroke.getKeyStroke(KeyEvent. VK_S,Event.CTRL_MASK));
E) JMenuItem save = new JMenuItem("Save"); save.enableShortcut(KeyEvent.CTRL_KEY + KeyEvent.S);

```
public void print(Printable p){
PrinterJob pj = PrinterJob.getPrinterJob();
PageFormat pf = pj.pageDialog(pj.defaultPage());
pj.setPrintable(p,pf);
// now get user to select properties and confirm print
}
```

Given the above sample code, which takes a Printable object and sets up a PrinterJob to print that object, what code would you use to allow the user to select properties for the print job, confirm, and then print?
A) p.print();
B) pj.printerDialog(pj.getDefaultPrinterOptions()); if(pj.printConfirmed()) p.print();
C) PrinterDialog pd = new PrinterDialog(); if(pd.show() == Dialog.OK) pj.print(p);
D) Properties props = pj.getUserOptions(); if(((Boolean)props.getValue(PrinterJob.CONFIRM)).equals(true)) p.print(pj);
E) if(pj.printDialog()) pj.print();
What is an RMI "stub"?
A) It acts as a client-side proxy to receive RMI calls and pass them to the server.
B) It receives RMI calls from the server to enable applet callbacks.
C) It is used by the client to find remote objects on a server.
D) It is the interface implemented by both the client and server objects.

E) It enables applets to make RMI calls without violating browser security.

Which of the following types of audio files does JDK 1.2 NOT support?

A) MIDI
B) RMF
C) AU
D) MP3
E) WAV

```
1: Date myDate = new Date();
2: DateFormat dateFormat = DateFormat.getDateInstance();
3: String myString = dateFormat.format(myDate);
4: System.out.println( "Today is " + myString );
```

How can you change line 2 above so that the date is displayed in the same format used in China?

A) DateFormat dateFormat = DateFormat.getDateInstance();
dateFormat.setLocale(java.util.Locale.CHINA);
B) DateFormat dateFormat = DateFormat.getDateInstance(
DateFormat.SHORT, java.util.Locale.CHINA);
C) java.util.Locale locale = java.util.Locale.getLocale("China");
DateFormat dateFormat = DateFormat.getDateInstance(locale);
D) DateFormat.setLocale(java.util.Locale.CHINA);
DateFormat dateFormat = DateFormat.getDateInstance();
E) java.util.Locale locale = java.util.Locale.getLocale(java.util.Locale.CHINA);
DateFormat dateFormat = DateFormat.getDateInstance(locale);

How would you implement a text field for entering a password in a Login dialog using Swing components?

A) By using a JTextField and call setEchoChar('*').
B) By using a JPasswordField.
C) By using a JTextField and adding an event handler for the KeyPressed event to echo a '*' character.
D) By creating your own class which extends JTextField and overrides the keyDown() method to echo a '*' character.
E) use a JTextComponent and call setTextDisplay(JTextComponent.PASSWORD).

Which aspect(s) of the bean is used to modify or retrieve the JavaBeans properties?

A) Accessor methods
B) Persistence mechanism
C) propertyManager() method (required in all beans)
D) Event adapters
E) Associated java.beans.PropertyChangeListener class methods

```
class A {
Object get(Object o) {
try {
return o.getClass().getName();
} catch (Exception e) {
return null;
}
}
}
```

Referring to the above, what is the expected output when the following code is executed?

```
A a = new A();
Vector v = new Vector();
System.out.println( a.get( v ) );
```

A) NullPointerException
B) A
C) Object
D) null
E) java.util.Vector

What does double-buffering in animations do?

A) It draws the next frame to an off-screen image object before displaying to eliminate flicker.
B) It prevents "hanging" using a MediaTracker object to ensure all frames are loaded before beginning.
C) It puts multiple frame in the same file and uses cropImage() to select the desired frame.
D) It uses a small image as wallpaper and a transparent image as the actual frame.
E) It draws each image to the screen twice to eliminate white spots.

```
class Class1 {
int total=0;
public static void main(String args[]) {
doIt();
}
void doIt() {
for(int i=0;i<5;i++) total += i;
System.out.println(total);
}
}
```

In reference to the above, how should the first line of method main() be changed to call doIt()?

A) Class1();
B) Class1 class1 = new Class1();
C) (new Class1()).doIt();
D) Class1().doIt();
E) No change is necessary.

In a CORBA application, how would a client create a new remote object on the server?

A) By calling an existing server object's "factory" method which would create a new object and return a reference to the client.
B) By using com.omg.CORBA.ORB.string_to_object() method to instantiate a new object from a stringified object reference.
C) By using a naming service, such as the tnameserv in JDK 1.2, to create the object.

D) By using the Remote Object Activation facility.

E) By using the server objects Home interface which provides methods for finding, creating, and deleting objects.

```
static {
System.loadLibrary("mNativeLib");
}
public int myMethod(int count) {
Additional code here
}
```

In reference to the above, if myMethod() is implemented in native code library "mNativeLib", how must its declaration be changed?

A) Replace myMethod()'s code with System.execute("mNativeLib", myMethod(count));.

B) Replace it with native myMethod(int count) {}.

C) Replace it with public System.nativeMethod("myMethod(int)");.

D) Move it into the static code block.

E) Replace it with public native int myMethod(int count);.

Which of the following is a requirement of a JavaBean?

A) Define a java.beans.PropertyEditor.

B) Use of the java.beans.PropertyChangeListener.

C) Provide a no-argument constructor.

D) Implement the java.beans.BeanInfo interface.

E) Extend java.beans.BeanDescriptor.

How would you disable tooltips in a Swing based Java application?

A) Tool tip support is determined by the underlying windowing system, and cannot be changed by a Java application.

B) UIManager.disableToolTips();

C) ToolTipManager.sharedInstance().setEnabled(false);

D) For each component in the application call: JComponent.setToolTipText(null);

E) SwingUtilities.setToolTipsEnabled(false);

Which of the following is NOT a valid java.lang.String declaration?

A) String myString = new String();

B) String cde = "cde";

C) String myString = new String(5);

D) char data[] = {'a','b','c'};
String str = new String(data);

E) String myString = new String("Hello");

Which of the following is true for untrusted applets running on most browsers?

A) Applets can connect to computers other than the codebase.

B) Applets can read to or write from files in the "/tmp" or "c:\tmp" directory.

C) Applets can change the ClassLoader.

D) Applets can retrieve the user's account name.

E) Applet frames display a warning.

In reference to the above, when you click on the button the applet changes background colors. Which method must it call to do this?

A) repaint()

B) shade()

C) draw()

D) setbgcolor()

E) restart()

In what way is it possible to circumvent applet security restrictions?

A) Install a different SecurityManager.

B) Attach a trusted digital signature to the downloaded applet JAR file.

C) Replace the default ClassLoader.

D) Connect to a ActiveX control located in the same web page.

E) There is no way to circumvent applet security restrictions.

```
1: class C extends Thread {
2: public void run() {
3: while(true) {
4: System.out.println("Hello World!");
5: try {
6: sleep(100);
7: } catch(Exception e) {}
8: }
9: }
10: public static void main(String[] s) {
11: C c = new C();
12: c.start();
13: }
14: }
```

What is the result of the program above?

A) A java.lang.StackOverflowException is thrown.
B) The CPU is used complete up by the infinite loop.
C) Nothing. Compiler error on line 6.
D) The system exits immediately with little or no output.
E) "Hello World!" is printed forever.

```
public class TestApplet extends JApplet{
public void init(){
JButton b1 = new JButton("one");
JButton b2 = new JButton("two");
JButton b3 = new JButton("three");
JButton b4 = new JButton("four");
JButton b5 = new JButton("five");
getContentPane().add(b1);
getContentPane().add(b2);
getContentPane().add(b3);
getContentPane().add(b4);
getContentPane().add(b5);
}
}
```

Given the above sample code of a JApplet, what would be displayed when the applet is run?

A) Only the button "five", occupying the entire area of the applet.
B) Five buttons, "one","two","three","four","five", in a column top to bottom.
C) Five buttons, "one","two","three","four","five", in a row left to right.
D) Only the button "one", occupying the entire area of the applet.
E) Five buttons, "one","two","three","four","five", randomly located within the applet.

```
package mypackage;
class MyException extends java.lang.Exception {
public MyException() { super(); }
public MyException( String s ) { super(s); }
}
```

Referring to the above, which code segment properly generates an exception of type "MyException"?

A) exception new MyException();

B) new Exception(MyException);
C) catch(MyException e) { }
D) throw new MyException("Error Occurred!");
E) MyException e = new MyException();
return e;

Sample Code

```
public void drawText(Graphics2D g, String text){
Font f = new Font("Century Gothic", Font.BOLD, 10);
g.setFont(f);
g.drawString(text, 20, 20);
}
```

Given the above sample code, what would happen if the "Century Gothic" font was not available?

A) The Font() constructor will fail causing f to be null, and the setFont() method will throw a NulPointerException.
B) The code will not compile, because only the following "logical" font names are supported in Java: Dialog, DialogInput, Monospaced, Serif, SansSerif, and Symbol.
C) A "default" Font object will be returned and this will be used.
D) The Font() constructor will throw an IllegalArgumentException.
E) "Century Gothic" is a built in Java font, which will always exist within the Java runtime environment.

Sample Code

```
1: Date myDate = new Date();
2: DateFormat dateFormat = DateFormat.getDateInstance();
3: String myString = dateFormat.format(myDate);
4: System.out.println( "Today is " + myString );
```

How can you change line 2 above so that the date is displayed in the same format used in China?

A) DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.SHORT, java.util.Locale.CHINA);
B) java.util.Locale locale = java.util.Locale.getLocale("China");
DateFormat dateFormat = DateFormat.getDateInstance(locale);
C) java.util.Locale locale = java.util.Locale.getLocale(java.util.Locale.CHINA);
DateFormat dateFormat = DateFormat.getDateInstance(locale);
D) DateFormat dateFormat = DateFormat.getDateInstance();
dateFormat.setLocale(java.util.Locale.CHINA);
E) DateFormat.setLocale(java.util.Locale.CHINA);
DateFormat dateFormat = DateFormat.getDateInstance();

Which Java construct maps to the CORBA IDL 'module' construct?

A)
interface
B)
package
C)
static variable
D)
class
E)
method

How would you disable tooltips in a Swing based Java application?

A) UIManager.disableToolTips();
B) SwingUtilities.setToolTipsEnabled(false);
C) For each component in the application call: JComponent.setToolTipText(null);
D) ToolTipManager.sharedInstance().setEnabled(false);

E) Tool tip support is determined by the underlying windowing system, and cannot be changed by a Java application.

Sample Code

```
void myMethod(int foo, int bar, String str) {
do {
if(foo <= bar) {
int i = 1;
System.out.println(str + ": " + i);
i++;
}
} while(i<10);
}
```

What is the error in the above code?

A) myMethod does not define a valid return type.

B) You cannot concatenate strings inside a call to System.out.println().

C) If foo is greater than bar, the "do" loop will never terminate.

D) The "while" statement should appear to the left of the "}" bracket.

E) The "while" statement does not have access to variable i.

Sample Code

```
class PrimeThread extends Thread {
long minPrime;
long result;
PrimeThread(long minPrime) {
this.minPrime = minPrime;
}
public void run() {
// compute primes larger than minPrime
// . . .
}
}
```

Referring to the above, which code segment could create and start the Prime Thread?

A) PrimeThread(143).run();

B) PrimeThread p = new PrimeThread();
p.run();

C) Thread p = new Thread(new PrimeThread(143));
p.run();

D) (new PrimeThread(143)).start();

E) Runnable r = new Runnable(PrimeThread(143));
r.start();

Sample Code

```
public double SquareRoot( double value ) throws ArithmeticException
{
if (value >= 0) return Math.sqrt( value );
else throw new ArithmeticException();
}
public double func(int x) {
double y = (double) x;
try {
y = SquareRoot( y );
}
catch(ArithmeticException e) { y = 0; }
finally { --y; }
return y;
}
```

Referring to the above, what value is returned when method func(4) is invoked?

A) -2.0
B) -1.0
C) 0
D) 1.0
E) 2.0

Sample Code

```
import java.awt.*;
class A extends Frame implements ActionListener,
WindowListener {
public A() {
Button button1 = new Button("Hello");
add(button1);
show();
}
public actionPerformed(ActionEvent e) {
// respond to mouse click
}
}
```

In reference to the above, what statement could be added to the constructor for class A to monitor for a mouse click on button1?

A) button1.addListener(ActionListener);
B) button1.addActionListener(this);
C) super.addWindowListener(button1);
D) this.addActionListener(button1);
E) this.addEventListener(button1);

Which method would be most efficient for inserting a new row into a table?

A) statement.insertRow()
B) statement.executeQuery()
C) statement.executeUpdate()
D) statement.execute()
E) statement.insertRecord()

Which statement correctly describes the support for "Drag and Drop" provided by JDK 1.2?

A) "Drag and Drop" is supported between Java objects in the same JVM, and between Java applications running in different JVMs.
B) "Drag and Drop" is not yet supported. The only supported data transfer method is "Cut and Paste".
C) "Drag and Drop" is only supported between Java objects running in the same JVM.
D) "Drag and Drop" is supported between Java objects in the same JVM, between Java applications running in different JVMs, and between a Java application and a native application.
E) "Drag and Drop" is supported between Java objects in the same JVM, between Java applications running in different JVMs, between a Java application and a native application, and between two native applications.

In JavaBeans, what is a "constrained" property?

A) It is a property that allows changes of its value to be vetoed by registered listeners.
B) It is read-only.
C) It causes the entire object to re-validate when it is modified.
D) It is tied to another internal (non-public) property.
E) It is static at runtime.

Sample Code

```
import java.rmi.*;
public class A {
```

```
public static void main(String args[]) {
int x=0, y=2;
anObject remoteObject;
try {
remoteObject = (anObject) Naming.lookup("rmi://h.com/B");
x = remoteObject.method1(y);
} catch(Exception e) {}
}
}
```

In reference to the above, which of the following is NOT a valid conclusion regarding "anObject"?

A) anObject contains method method1() that takes an int parameter.
B) anObject extends java.rmi.Remote.
C) anObject is implemented by class B.
D) anObject responds to toString().
E) anObject is an interface.

Widget
Choice A
Choice B
Choice C
Choice D
Choice E

The widget shown above is similar to which standard java.awt class?

A) List
B) Select
C) PullMenu
D) Choice
E) Menu

Which of the following commands starts the Java IDL name service on port 1234?

A) idlregistry -IDLInitialPort 1234
B) nameserver -InitialPort 1234
C) cosnaming -COSInitialPort 1234
D) nameservice 1234
E) tnameserv -ORBInitialPort 1234

Sample Code

```
java.awt.Panel panel[] = new java.awt.Panel[3];
panel[0] = new java.awt.Panel();
panel[0].setSize(500,500);
panel[0].setLayout( new java.awt.BorderLayout() );
panel[1] = new java.awt.Panel();
panel[1].setLayout( new java.awt.FlowLayout(java.awt.FlowLayout.RIGHT) );
panel[0].add( panel[1], "South" );
panel[2] = new java.awt.Panel();
panel[2].setSize(400,400);
panel[0].add( panel[2], "Center" );
java.awt.Button button1 = new java.awt.Button("Click Me!");
panel[1].add(button1);
```

Where will button1 be positioned on panel[0] as shown above?

A) In the lower right-hand corner.
B) In the upper right-hand corner.
C) In the upper left-hand corner.
D) In the lower left-hand corner.
E) Centered horizontally at the top.

Sample Code

```
public drawThickLine(Graphics2D g, x1, y1, x2, y2){
```

```
// set line thickness here
g.drawLine(x1,y1,x2,y2);
}
```

Given the above sample code, what code would you insert at the indicated point to make the method draw a line that is ten pixels wide?

A) Stroke s = new Stroke(); s.setWidth(10); g.setStroke(s);

B) g.setLineWidth(10);

C) g.setStroke(new BasicStroke(10.0f));

D) g.setPen(new DefaultPen(10));

E) g.setBrush(new Brush(10.0));

Sample Code

```
class Employee{
protected String name, salary;
public Employee() {}
public Employee(String aName, String aSalary) {
name=aName; salary=aSalary;
}
public String toString(){ return (name + ", " + salary); }
}
class Manager extends Employee{
protected String car;
public Manager(String aName, String aSalary, String aCar) {
car = aCar;
super( aName, aSalary );
}
public String toString(){ return (name + ", " + salary + ", " + car); }
}
public class EmployeeTest{
public static void main( String[] args ){
Employee[] employees = new Employee[3];
employees[0] = new Employee("Smith","$25,000");
employees[1] = new Employee("Jones","$35,000");
employees[2] = new Manager("Wilson","$45,000","BMW");
for(int i=0; i<employees.length; i++)
System.out.println(employees[i]);
}
}
```

Given the above sample code, what will be the result when the code is run?

A) The code will NOT compile, because "super(aName,aSalary);" must be the first line in the Manager constructor.

B) Smith, $25,000 Jones, $35,000 Wilson, $45,000, BMW

C) The code will NOT compile, because the Manager class cannot access name and salary from the Employee class.

D) Smith, $25,000 Jones, $35,000 Wilson, $45,000

E) The code will NOT compile, because System.out.println() does not take an Employee parameter.

Which of the following is NOT a valid java.lang.String declaration?

A) char data[] = {'a','b','c'};
String str = new String(data);

B) String myString = new String();

C) String myString = new String("Hello");

D) String cde = "cde";

E) String myString = new String(5);

Sample Code

```
public class TimerTest implements ActionListener{
```

```
public static void main(String[] args){
TimerTest test = new TimerTest();
javax.swing.Timer timer = new javax.swing.Timer(100,test);
}
public void actionPerformed(ActionEvent ev){
System.out.println("Timer ticked.");
}
}
```

Given the above sample code, what would be the result when the code is run?

A) The words "Timer ticked." will be printed to the console once, after 100 milliseconds.
B) The words "Timer ticked." will be printed to the console every 100 microseconds.
C) Nothing, because the Timer is not started.
D) The words "Timer ticked." will be printed to the console every 100 seconds.
E) The words "Timer ticked." will be printed to the console every 100 milliseconds.

Which command-line tool would you use to create public/private key pairs?

A) policytool
B) keytool
C) keymanager
D) jarsigner
E) javakey

What is NOT a typical feature of visual JavaBeans?

A) Events, so that a simple communication metaphor can be used to connect up beans.
B) Customization, so that when using an application builder a user can customize the appearance and behavior of a bean.
C) Persistence, so that a bean can be customized in an application builder and then have its customized state saved away and reloaded later.
D) Properties, both for customization and for programmatic use.
E) Distributed framework, so the visual component resides on the client, while the logic resides on the server.

Sample Code

```
class Animal{
public Animal(int numLegs){}
}
class Dog extends Animal{
public void bark(){ System.out.println("woof!"); }
}
public class DogTest{
public static void main( String[] args ){
Dog rover = new Dog();
rover.bark();
}
}
```

What will be the result when the above sample code is compiled and run?

A) The code will compile, but will cause a runtime error because the JVM will not be able to load classes Animal and Dog.
B) The code will not compile because the class Dog has no constructor Dog().
C) The code will not compile because the class Animal has no constructor Animal().
D) "woof!" will be printed to the console.
E) The code will not compile because the classes Animal and Dog are not declared as public.

Sample Code

```
module Test{
interface Tester{
long doTest();
};
};
```

Given the above sample code, which JDK 1.2 tool would you use to generate the corresponding Java interface, stub, skeleton, helper, and holder classes?

A) rmic

B) tnameserv

C) idlc

D) javac

E) idltojava

What class should be used to represent the U.S. Mountain time zone in date/time calculations?

A) java.util.Locale

B) java.util.Date

C) java.util.SimpleTimeZone

D) java.util.GregorianCalendar

E) java.util.Calendar

Sample Code

```
<import java.io.*;
Additional code here
FileInputStream f = new FileInputStream("store");
ObjectInputStream in = new ObjectInputStream(f);
Object obj = in.readObject();
```

In reference to the above, what additional code will produce the type of object represented by "obj"?

A) obj.toString();

B) String name;

```
String all[] = {"String","StringBuffer","Array", others here};
for(int i=0;i<all.length;i++)
if(name = all[i].equals(Class.instanceOf(all[i]))) break;
```

C) obj.getClass();

D) new Class.getName(obj);

E) Classloader.getInstance(obj);

What class in java.security can be subclassed to represent entities that can be authenticated using their public keys?

A) DigitalSignature

B) Signer

C) Signature

D) Identity

E) KeyPair

What happens if a parameter is passed to a remote object (using RMI) which does NOT implement Remote or Serializable?

A) It is passed by reference.

B) It is passed by value.

C) A java.lang.NullPointerException is thrown.

D) A java.rmi.MarshallException is thrown.

E) The code will not compile.

Which of the following do you NOT have to develop when creating an Entity EJB?

A) Entity context

B) Remote interface

C) Primary Key

D) Bean implementation

E) Home interface

Which interface would you implement if you want your class to handle its own serialization?
A) Externalizable
B) Streamable
C) Cloneable
D) ObjectOutput
E) Serializable

Which of the following is a requirement of a JavaBean?
A) Provide a no-argument constructor.
B) Define a java.beans.PropertyEditor.
C) Extend java.beans.BeanDescriptor.
D) Implement the java.beans.BeanInfo interface.
E) Use of the java.beans.PropertyChangeListener.

What is the primary purpose of the RMI/IIOP extension?
A) To allow RMI objects to communicate with Servlets over HTTP.
B) To allow RMI objects to communicate with CORBA objects.
C) To allow RMI objects to communicate with DCOM objects.
D) To allow RMI objects to communicate through a firewall.
E) To allow RMI objects to communicate with Enterprise JavaBeans.

Sample Code

```
class A {
int method1(int i) {
return i*2;
}
int method1(float f) {
int i = (int) f;
return i*2;
}
}
```

In reference to the above, will attempting to compile class A pass or fail?
A) It will fail because floats cannot be cast to integers.
B) It will fail because there is no inheritance information.
C) It will fail because there are duplicate methods.
D) It will fail because there is no main method.
E) Succeed.

Sample Code

```
public class ArrayTest{
public static void main(String[] args){
// insert code here
}
}
```

Given the above sample code, what would you insert to print out a list of the command-line arguments?
A) for(int i=0; i<sizeof(args); i++) System.out.println(args[i]);
B) for(int i=0; i<args.length(); i++) System.out.println(args[i]);
C) for(int i=0; i<args.length; i++) System.out.println(args[i]);
D) for(int i=0; i<args.size(); i++) System.out.println(args[i]);
E) for(int i=0; i<args.getLength(); i++) System.out.println(args[i]);

Sample Code

```
public void parseUserList( List users ){
Iterator iterator = users.iterator();
while( iterator.hasNext() ){
String user = iterator.next();
if( user.equals("Smith") )
iterator.remove();
}
}
```

The above sample code is a method that takes a list of usernames and parses them to remove any users named "Smith". What is wrong with the code?

A) next() returns an Object, NOT a String, so you have to cast the returned value to a String.

B) Iterator does not support removing elements from the underlying List; you should use users.remove("Smith").

C) Iterator does not have a method called next(); it should be nextElement().

D) List does not have a method called iterator(); it should be getIterator().

E) Iterator does not have a method called hasNext(); it should be hasMoreElements().

Which of the following services is NOT provided by Java RMI?

A) Remote object activation

B) Naming service

C)Distributed garbage collection

D) Dynamic class loading

E) Distributed transaction management

Sample Code

```
public void createTempFiles(String d) {
File f = new File(d);
f.mkdirs();
// more code here ...
}
```

What is the result when the following statement is invoked on the code above on a Unix platform?

```
createTempFiles( "/tmp/myfiles/_3214" );
```

A) The "myfiles" directory is created in the "/tmp" directory.

B) The directory "/tmp/myfiles/_3214" is created if it doesn't already exist.

C) A java.io.DirectoryNotCreatedException is thrown.

D) The file "_3214" is created in the "/tmp/myfiles" directory.

E) A java.lang.SecurityException is thrown.

Sample Code

```
int f = 2;
int g = 5;
double h;
h = 3+f/g+2;
```

What is the expected value for h after execution of the above code?

A) 0.71

B) 5.0

C) 5.2

D) 5.4

E) 6

How can you ensure compatibility of persistent object formats between different class versions?

A) Implement java.io.Serializable.

B) Override the default ObjectInputStream and ObjectOutputStream.

C) Implement java.io.Serilizable, add a serial version UID, and write your own readObject() method to handle the different versions.

D) Make the whole class transient.

E) Add "transient" variables.

Sample Code

```
class Super{
protected int getX( int a ){ return 10*a; }
}
```

```
class Sub extends Super{
protected int getX( int b ){ return 5*b; }
}
public class Test{
public static void main( String[] args ){
Super s = new Sub();
System.out.println( s.getX(1) );
}
}
```

What will be printed when the above code is run?

A) Nothing, because Sub has no constructor.
B) 5
C) 10
D) 50
E) The code will not compile because you cannot override a protected method.

Sample Code

```
class Class1 {
int total=0;
public static void main(String args[]) {
doIt();
}
void doIt() {
for(int i=0;i<5;i++) total += i;
System.out.println(total);
}
}
```

In reference to the above, how should the first line of method main() be changed to call doIt()?

A) Class1();
B) Class1 class1 = new Class1();
C) (new Class1()).doIt();
D) Class1().doIt();
E) No change is necessary.

What is the primary purpose of the RMI/IIOP extension?

A) To allow RMI objects to communicate with DCOM objects.
B) To allow RMI objects to communicate through a firewall.
C) To allow RMI objects to communicate with Servlets over HTTP.
D) To allow RMI objects to communicate with Enterprise JavaBeans.
E) To allow RMI objects to communicate with CORBA objects.

How is an applet started by a browser?

A) The browser calls repaint(), which ends with a call to run().
B) Applets are not executed. They are displayed when the browser calls paint().
C) The browser loading the applet calls run().
D) The browser calls init() and then start().
E) The browser calls init().

Which method is invoked each time a Servlet is invoked?

A) init()
B) process()
C) start()
D) service()
E) run()

Sample Code

```
public class TestServlet extends HttpServlet{
public void doGet(HttpServletRequest request, HttpServletResponse response){
// insert code here
```

}
}
Given the above sample servlet code, what code fragment could you insert to print out the name and value of the cookies passed from the client that initiated this reqest?

A) String[] names = request.getSession().getValueNames(); for(int i=0; i<names.length; i++){ String cookieValue = request.getSession().getValue(names[i]); System.out.println(names[i]+","+cookieValue); }

B) The Servlet API does NOT support the use of cookies, since they may not be available in all browsers.

C) Cookie[] cookies = request.getCookies(); for(int i=0; i<cookies.length; i++) System.out.println(cookies[i].getName()+","+cookies[i].getValue());

D) The Servlet API only supports the use of cookies for maintaining a sessionID on the client.

E) Enumeration cookies = response.getCookies(); while(cookies.hasMoreElements()){ Cookie cookie = (Cookie)cookies.nextElement(); System.out.println(cookie.getName()+","+cookie.getValue()); }

What happens when you try to open a scrollable ResultSet on a JDBC driver that does NOT support scrollable ResultSets?

A) The standard JDBC classes handle scrolling for you.

B) A SQLException is thrown and no ResultSet is created.

C) All JDBC drivers support scrollable ResultSets.

D) A Java 1.2 JVM will not allow you to load a driver that does not support scrollable ResultSets.

E) A SQLWarning should be issued on the Connection and a non-scrollable ResultSet is returned.

Widget
Choice A
Choice B
Choice C
Choice D
Choice E

The widget shown above is similar to which standard java.awt class?

A) Menu

B) List

C) Choice

D) Select

E) PullMenu

Which statement correctly describes the support for "Drag and Drop" provided by JDK 1.2?

A) "Drag and Drop" is not yet supported. The only supported data transfer method is "Cut and Paste".

B) "Drag and Drop" is supported between Java objects in the same JVM, between Java applications running in different JVMs, and between a Java application and a native application.

C) "Drag and Drop" is only supported between Java objects running in the same JVM.

D) "Drag and Drop" is supported between Java objects in the same JVM, and between Java applications running in different JVMs.

E) "Drag and Drop" is supported between Java objects in the same JVM, between Java applications running in different JVMs, between a Java application and a native application, and between two native applications.

What is NOT a typical feature of visual JavaBeans?

A) Events, so that a simple communication metaphor can be used to connect up beans.
B) Distributed framework, so the visual component resides on the client, while the logic resides on the server.
C) Properties, both for customization and for programmatic use.
D) Persistence, so that a bean can be customized in an application builder and then have its customized state saved away and reloaded later.
E) Customization, so that when using an application builder a user can customize the appearance and behavior of a bean.

Sample Code

```
public void paint(Graphics g){
Graphics2D g2d = (Graphics2D) g;
// turn off antialiasing here
// display text
}
```

Given the above sample code, how would you turn off text antialiasing before you display the text?

A) UIManager.setRenderingHints(Graphics2D.TEX_ANTIALIASING_ENABLED, false);
B) SwingUtilities.setAntialisingEnabled(false);
C) g2d.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING, RenderingHints.VALUE_TEXT_ANTIALIAS_OFF);
D) RenderingHints hint = new RenderingHints(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_OFF); g2d.addRenderingHint(rh);
E) You CANNOT control whether antialiasing is enabled. It depends on the native windowing system.

Sample Code

```
module test{
interface Test{
long doTest();
};
};
```

Given the above sample IDL file, Test.idl, which of the following files is NOT generated by running "idltojava Test.idl"?

A) TestHelper.java
B) _TestStub.java
C) Test.java
D) TestServer.java
E) TestHolder.java

Sample Code

```
class A{ }
class B extends A{ }
class C extends A{ }
class Test{
public static void main(String[] args){
A a = new A();
B b = new B();
C c = new C();
// assignment here
}
}
```

Given the above sample code, which of the following assignments would be legal?

A) a=b
B) b=a

C) b=c
D) None of the above.
E) c=a
Sample Code

```
public class Test{
public static void main(String[] args){
int x=5, y=7;
swap(x,y);
System.out.println("x = " + x + ", y = " + y );
}
static void swap(int a, int b){
int c = a;
a = b;
b = c;
}
}
```

Given the above sample code, what will be printed when the Test program is run?
A) x = 12, y = 12
B) x = 7, y = 5
C) x = 0, y = 0
D) x = 5, y = 7
E) The code will not compile, because the swap() method is not visible from main().

```
class Class1 {
int total=0;
public static void main(String args[]) {
(new Class1()).doIt();
}
void doIt() {
for(int i=0;i<5;i++) total += i;
System.out.println(total);
}
}
```

Which of the following Swing components is NOT a lightweight component?
A) JButton
B) JPanel
C) JFrame
D) JList
E) JRootPane
Which Java construct maps to the CORBA IDL 'module' construct?
A) class
B) static variable
C) method
D) interface
E) package
Which of the following is a limitation of subclassing the Thread class?
A) It cannot have any static methods in the class.
B) It must catch the ThreadDeath exception.
C) It must implement the Threadable interface.
D) It cannot subclass any other class.
E) It must declare the class final.
Sample Code

```
import java.rmi.*;
public class A {
public static void main(String args[]) {
```

```
int x=0, y=2;
anObject remoteObject;
try {
remoteObject = (anObject) Naming.lookup("rmi://h.com/B");
x = remoteObject.method1(y);
} catch(Exception e) {}
}
}
```

In reference to the above, which of the following is NOT a valid conclusion regarding "anObject"?

A) anObject responds to toString().
B) anObject is an interface.
C) anObject contains method method1() that takes an int parameter.
D) anObject is implemented by class B.
E) anObject extends java.rmi.Remote.

Sample Code

```
void shifter(int[] array, int arrayLength) {
for(int i=1;i<arrayLength;i++) {
array[i-1] = array[i];
}
}
```

How could you prevent other threads from changing data in "array" from the code above while method shifter() is executing?

A) Declare "array" as static.
B) Declare shifter as synchronized.
C) Call wait() before executing the for-loop.
D) Add "synchronized" to the "array[i-1] = array[i];" statement.
E) Enclose all code in all classes changing "array" in a synchronized (array) code block.

Sample Code

```
public class Applet2D extends JApplet{
public void paint(Graphics g){
Graphics2D g2d = // *** insert code here ***
// further graphics code ...
}
}
```

Given the above sample code, how would you obtain a reference to a Graphics2D object to allow you to use the advanced graphics functions supported by the Java 2D API?

A) Graphics2D g2d = UIManager.getGraphics2D();
B) Graphics2D g2d = g.getGraphics2D();
C) Graphics2D g2d = new Graphics2D(g);
D) Graphics2D g2d = SwingUtilities.getGraphics2D();
E) Graphics2D g2d = (Graphics2D) g;

What is Java serialization?

A) Marshaling and unmarshaling of remote objects.
B) Remote method invocation.
C) The ability transform the state of an object into bits and resurrect a copy of the object from those bits.
D) Distributed persistence.
E) The ability to examine the encapsulated data of a class.

What are the two main types of persistence for entity EJBs?

A) Bean-managed and Container-managed
B) Application-managed and Database-managed
C) Client-managed and Server-managed

D) Context-managed and Session-managed

E) Applet-managed and Servlet-managed

Which of the following actions is NOT required when converting a standard RMI object to work with RMI-IIOP?

A) Add code to initialize the ORB: ORB orb = org.omg.CORBA.ORB.init();

B) Replace any casting of Remote objects to Java types with the javax.rmi.PortableRemoteObject.narrow() method calls.

C) Replace any code that uses java.rmi.Naming with JNDI code to talk to the CORBA Naming Service.

D) Change the class to extend javax.rmi.PortableRemoteObject instead of java.rmi.server.UnicastRemoteObject.

E) Use rmic with the -iiop option to produce stub and tie classes.

Which aspect(s) of the bean is used to modify or retrieve the JavaBeans properties?

A) Event adapters

B) Accessor methods

C) propertyManager() method (required in all beans)

D) Persistence mechanism

E) Associated java.beans.PropertyChangeListener class methods

What prevents a malicious applet from loading its own SecurityManager, giving itself unlimited access rights?

A) The browser loads a SecurityManager before loading applets and only one is allowed.

B) SecurityManager cannot be loaded from a remote source.

C) SecurityManagers cannot be loaded unless the applet is digitally signed.

D) The applet needs private key of the browser to load a new SecurityManager.

E) The ClassLoader will deny loading a new SecurityManager.

Which of the following is NOT typically found in a digital certificate?

A) private key of the signer

B) serial number

C) organization of the signing entity

D) registered name of the signer

E) expiration date

Sample Code

```
class B extends A {
int flag = 0;
public int getFlag() {
return flag;
}
protected void setFlag(int newSetting) {
flag = newSetting;
}
}
```

Which statement describes the relationship between classes A and B in the code above?

A) The source code of B must be in the same file as A.

B) B is a subclass of A.

C) B is a superclass of A.

D) B belongs to the same package as A.

E) Instances of A have access to all of the methods defined by B.

Which method can an untrusted applet call without necessarily throwing a security exception?

A) new FileOutputStream("/tmp/myTemp.out");

B) System.getProperty(...)

C) System.exit(...)

D) ClassLoader.getSystemClassLoader()

E) Runtime.exec(...)

Sample Screen

Object 1 Here

Object 2 Here Object 3 Here Object 4 Here

Object 5 Here

To display five objects as shown above using all available screen space, what Layout Manager would be easiest?

A) CardLayout

B) FlowLayout

C) BorderLayout

D) null

E) GridLayout

Sample Code

```
String sql = "SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEE WHERE ID='123'";
Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATEABLE);
ResultSet rs = st.executeQuery(sql);
rs.first();
rs.updateString(1,"Tom");
rs.updateString(2,"Smith");
```

Given the above sample code, what would you insert after the last line in order to write the changes to the database?

A) rs.updateRow();

B) rs.close();

C) st.saveChanges();

D) rs.next();

E) con.commit();

Sample Code

```
public interface A {
int myCount = 10;
void method1(int i);
int method2(float f);
}
-- New File --
class B implements A {
}
```

In reference to the above, to implement A, what must class B do?

A) Class B must be abstract.

B) Class B must declare static versions of method1() and method2().

C) Class B must include non-abstract versions of method1() and method2(), or be declared as abstract itself.

D) Class B must be in the same package as A.

E) Class B must override variable myCount.

Sample Code

```
static Object get() {
return value;
}
```

Referring to the above, what could the declaration of "value" look like?

A) private static java.net.Socket value;

B) int value = 3;

C) public final Vector value;

D) private Object value = new Object();

E) public static int value = 10;

Sample Code

```
static Object get() {
```

```
return value;
}
```

Referring to the above, what could the declaration of "value" look like?

A) private static java.net.Socket value;

B) int value = 3;

C) public final Vector value;

D) private Object value = new Object();

E) public static int value = 10;

Sample Code

```
class A {
int i, j, k;
public A(int ii) { i = ii; }
public A() {
k = 1;
}
}
```

What code will instantiate an object of class A in the code above?

A) A(3) a;

B) A a = new A(4,8);

C) A a = new A(3);

D) A a = new A(3.3);

E) new A(this);

How can you know when a user closes a window so you can perform cleanup?

A) By creating an inner class called WindowListener, extending WindowEventHandler, then putting cleanup code in a processEvent(Event e) method.

B) By placing cleanup code in a try - catch block that catches a FrameClosingException.

C) By implementing WindowListener and putting cleanup code in a public void windowClosing(WindowEvent e) method.

D) By implementing FrameListener and placing cleanup code in a frameClosing(FrameEvent e) method.

E) Java does cleanup automatically when a window is closed.

```
String url=new String("http://www.tek.com");
```

How could you retrieve the content of the above URL?

A) Object content = new URLConnection(url).getContent();

B) String content = new URLConnection(url).collect();

C) String content = new URLHttp(url).getString();

D) Object content = new URL(url).getContent();

E) Socket content = new Socket(new URL(url)).collect();

Which class could you use to store a set of <key,value> pairs which are sorted by the keys?

A) TreeMap

B) HashMap

C) TreeSet

D) Hashtable

E) LinkedList

Sample Code

```
public class ArrayTest{
public static void main(String[] args){
// insert code here
}
}
```

Given the above sample code, what would you insert to print out a list of the command-line arguments?

A) for(int i=0; i<args.length(); i++) System.out.println(args[i]);

B) for(int i=0; i<args.getLength(); i++) System.out.println(args[i]);

C) for(int i=0; i<args.size(); i++) System.out.println(args[i]);

D) for(int i=0; i<args.length; i++) System.out.println(args[i]);

E) for(int i=0; i<sizeof(args); i++) System.out.println(args[i]);

Which Java API would you use to look up an Enterprise JavaBean (EJB) component?

A) JDBC

B) JMS

C) JNDI

D) Jini

E) JMAPI

Sample Code

```
public class CustomComponent extends JComponent implements Printable{
public int print(Graphics g, PageFormat pf, int pageNo){
// cast Graphics to Graphics2D
Graphics2D g2d = (Graphics2D) g;
// account for margins specified by PageFormat
g2d.translate(pf.getImageableX(),pf.getImageableY());
// call paint to print component
this.paint(g2d);
// tell the PrinterJob that the page was printed
return Printable.PAGE_EXISTS;
}
}
```

The above sample code shows part of the code for a "printable" component. What is wrong with the code?

A) You CANNOT pass a Graphics2D object to paint(); it takes a Graphics object.

B) You should first check if pageNo>=0 and return Printable.NO_SUCH_PAGE, to end the PrintJob, if not.

C) You should cast the Graphics to a PrinterGraphics object, NOT a Graphics2D.

D) PageFormat.getImageableX() and PageFormat.getImageableY() both return doubles, but Graphics2D.translate(x,y) expects int parameters.

E) You CANNOT cast a Graphics object to a Graphics2D.

Sample Code

```
public double SquareRoot( double value ) throws ArithmeticException
{
if (value >= 0) return Math.sqrt( value );
else throw new ArithmeticException();
}
public double func(int x) {
double y = (double) x;
try {
y = SquareRoot( y );
}
catch(ArithmeticException e) { y = 0; }
finally { --y; }
return y;
}
```

Referring to the above, what value is returned when method func(4) is invoked?

A) -2.0

B) -1.0

C) 0

D) 1.0

E) 2.0

Sample Code

```
int i=0;
float value = (float) 1.2;
String str = new String("Hello World");
boolean flag = false;
```

Which is a valid use of the variable "flag" in the sample code above?

A) i = value*flag;

B) if(!flag == i) System.out.println(str);

C) flag += i;

D) !flag = true;

E) if(flag) flag=false;

Sample Code

```
int h = 7;
float f = 3f;
float result = h%f;
```

In reference to the above, what is the value of "result" after execution?

A) 0

B) 1.0

C) 2.0

D) 2.33

E) 7.0

Sample Code

```
import java.lang.reflect.Constructor;
Constructor con;
con=new Constructor(Class.forName("Object"));
```

What is wrong with the above code?

A) Constructor is static and need not be instantiated.

B) Constructor has a private constructor, inaccessible to the code above.

C) Constructor objects can only be created for classes in the default package.

D) Constructor is abstract and cannot be instantiated.

E) Objects cannot be created for Object objects.

Which method would be most efficient for inserting a new row into a table?

A) statement.insertRow()

B) statement.executeQuery()

C) statement.execute()

D) statement.executeUpdate()

E) statement.insertRecord()

Sample Code

```
public class Test{
public static void main(String[] args){
StringBuffer[] messages = new StringBuffer[5];
messages[0].append("Hello world!");
System.out.println("First message is " + messages[0]);
}
}
```

What will be the output when the above sample code is run?

A) First message is null.

B) A NullPointerException will be thrown.

C) First message is Hello World!

D) The code would not compile.

E) An ArrayIndexOutOfBounds will be thrown.

In JavaIDL programming, the NamingContext allows you to bind objects to the naming service and look up objects. Which code fragment correctly gets a reference to the NamingContext (given that 'orb' is a reference to the ORB)?

A) org.omg.CORBA.ObjectHolder oh = orb.lookup_reference("NameService");
NamingContext nc = ObjectHelper.toNamingContext(oh);

B) org.omg.CORBA.Any any = orb.resolve_name_service("NameService");
NamingContext nc = NamingContextHolder.convert(any);
C) java.lang.Object o = orb.lookup("NameService"); NamingContext nc =
(NamingContext) o;
D) NamingContext nc = orb.getNamingContext("NameService");
E) org.omg.CORBA.Object o = orb.resolve_initial_references("NameService");
NamingContext nc = NamingContextHelper.narrow(o);

Which class could you use to store a set of <key,value> pairs which are sorted by the keys?

A) TreeSet
B) LinkedList
C) TreeMap
D) Hashtable
E) HashMap

Sample Code

```
public final class B extends A {
// ...
}
```

What is the meaning of "final" in the declaration above?

A) B cannot be subclassed.
B) This is the final version class A, which should be used if multiple versions are present.
C) Class A is an abstract class.
D) Class A does not need to be instantiated to invoke its methods.
E) None of class A's methods can be overridden.

Sample Code

```
class A{ }
class B extends A{ }
class C extends A{ }
class Test{
public static void main(String[] args){
A a = new A();
B b = new B();
C c = new C();
// assignment here
}
}
```

Given the above sample code, which of the following assignments would be legal?

A) b=c
B) a=b
C) c=a
D) None of the above.
E) b=a

When are constructors invoked?

A) When the object needs garbage collection.
B) When a superclass object is instantiated.
C) When a new instance of a class is instantiated.
D) When any method is invoked on an object.
E) When the java virtual machine begins garbage collection.

When designing a custom GUI component, which interface should you implement to provide support for assistive technologies such as screen magnifiers and speech recognition?

A) You do NOT have to implement an interface; simply have your component extend JComponent to inherit the functionality.
B) Accessible

C) Externalizable
D) Observer
E) InputMethodListener
Applet

In reference to the above, when you click on the button the applet changes background colors. Which method must it call to do this?

A) restart()
B) shade()
C) draw()
D) repaint()
E) setbgcolor()

Sample Code

```
public void parseUserList( List users ){
Iterator iterator = users.iterator();
while( iterator.hasNext() ){
String user = iterator.next();
if( user.equals("Smith") )
iterator.remove();
}
}
```

The above sample code is a method that takes a list of usernames and parses them to remove any users named "Smith". What is wrong with the code?

A) next() returns an Object, NOT a String, so you have to cast the returned value to a String.
B) Iterator does not have a method called hasNext(); it should be hasMoreElements().
C) Iterator does not support removing elements from the underlying List; you should use users.remove("Smith").
D) Iterator does not have a method called next(); it should be nextElement().
E) List does not have a method called iterator(); it should be getIterator().

Sample Code

```
class Animal{
public Animal(int numLegs){}
}
class Dog extends Animal{
public void bark(){ System.out.println("woof!"); }
}
public class DogTest{
public static void main( String[] args ){
Dog rover = new Dog();
rover.bark();
}
}
```

What will be the result when the above sample code is compiled and run?

A) The code will compile, but will cause a runtime error because the JVM will not be able to load classes Animal and Dog.
B) "woof!" will be printed to the console.
C) The code will not compile because the class Animal has no constructor Animal().
D) The code will not compile because the class Dog has no constructor Dog().
E) The code will not compile because the classes Animal and Dog are not declared as public.

What is the primary purpose of the RMI/IIOP extension?

A) To allow RMI objects to communicate with Servlets over HTTP.
B) To allow RMI objects to communicate with CORBA objects.

C) To allow RMI objects to communicate through a firewall.
D) To allow RMI objects to communicate with Enterprise JavaBeans.
E) To allow RMI objects to communicate with DCOM objects.
Sample Code

```
public class B extends A {
int xC,yC,k;
void move(int x) {
xC = x;
}
}
```

In reference to the above, what must you change to B so that it can be run in its own thread?
A) Implement Threadable.
B) Subclass ThreadGroup.
C) Make the class public, implement Runnable, and add a main() method.
D) Instantiate a new Thread.
E) Implement Runnable and add a "public void run()" method.
How are multiple requests for a servlet handled, assuming that the servlet does NOT implement SingleThreadModel?
A) If the servlet is busy handling a request when another request is received, an HTTP error 503 (service unavailable) is returned.
B) Only one request at a time is processed, and the others are placed in a queue.
C) A thread is allocated to execute the service method of the servlet.
D) It is up to the servlet programmer to determine how multiple requests are handled.
E) A new instance of the servlet is created to handle each request.
Sample Code

```
static Object get() {
return value;
}
```

Referring to the above, what could the declaration of "value" look like?
A) private static java.net.Socket value;
B) int value = 3;
C) public static int value = 10;
D) public final Vector value;
E) private Object value = new Object();
Sample Code

```
for(int i=0;i<=22;) {
if(i<= 10) {
int j= 2 + i;
i++;
// /* let us know */ //
System.out.println("i: " + i + " j: " + j);
}
}
```

What is the error in the above code?
A) The comment line is not formatted correctly.
B) You cannot print integer values without converting them to strings.
C) The loop will never terminate.
D) Variable j is referenced outside its scope.
E) You cannot declare variables inside a for-loop.
Sample Code

```
package mypackage;
class MyException extends java.lang.Exception {
public MyException() { super(); }
public MyException( String s ) { super(s); }
}
```

Referring to the above, which code segment properly generates an exception of type "MyException"?

A) MyException e = new MyException();
return e;

B) throw new MyException("Error Occurred!");

C) exception new MyException();

D) new Exception(MyException);

E) catch(MyException e) { }

Which of the following does the java.security package NOT address?

A) access control lists

B) cookies

C) key management

D) digital signatures

E) message digests

Sample Code

```
public void actionPerformed(ActionEvent e) {
}
```

What interface does the class containing the above code or a superclass most likely implement?

A) WindowListener

B) ActionAdapter

C) AWTEvent

D) ActionListener

E) ComponentEvent

In what way is it possible to circumvent applet security restrictions?

A) Replace the default ClassLoader.

B) Attach a trusted digital signature to the downloaded applet JAR file.

C) Connect to a ActiveX control located in the same web page.

D) There is no way to circumvent applet security restrictions.

E) Install a different SecurityManager.

Sample Code

```
class A {
public static void main(String args[]) {
int x=2,y=3,z=7;
if(x!=y) {
x += 4;
x *= (y-1);
z %= y;
}
}
}
```

What are ending values of variables x, y, and z after running the above code?

A) z=6, y=3, z=3

B) x=12, y=3, z=7

C) x=12, y=3, z=1

D) x=4, y=2, z=2

E) x=6, y=2, z=3

How would you disable tooltips in a Swing based Java application?

A) SwingUtilities.setToolTipsEnabled(false);

B) For each component in the application call: JComponent.setToolTipText(null);

C) ToolTipManager.sharedInstance().setEnabled(false);

D) UIManager.disableToolTips();

E) Tool tip support is determined by the underlying windowing system, and cannot be changed by a Java application.

Sample Code

```
import java.sql.*;
. . . Other Code Here. . .
Class.forName("any.sql.AsqlDriver");
String u = "jdbc:asql://mhost:1234/biz";
Connection c = DriverManager.getConnection(u);
```

In reference to the above, what does the AsqlDriver class do?

A) It registers a driver with JDBC when it is loaded.
B) It is written 100% in java.
C) It can only be accessed through a Class object.
D) It is monitoring port 1234 on mhost.
E) It does not use ODBC.

Which command starts the RMI Activation daemon, which manages remote object activation?

A) rmiad
B) rmiregistry
C) rmia
D) rmic
E) rmid

Sample Code

```
public void printIt(String txt) {
StringTokenizer st = new StringTokenizer(txt);
while (st.hasMoreTokens()) {
System.out.println(st.nextToken());
}
}
```

Referring to the above, what is the result when the following statement is invoked?
printIt(" Hello\n World\t!");

A) The following is outputted:
HelloWorld!
B) The following is outputted:
Hello
World
!
C) The following is outputted:
Hello
World
!
D) The following is outputted:
Hello
World!
E) java.util.NoSuchElementException is thrown.

Why would you use a WeakReference object?

A) It is useful for mapping to database types that do not easily map to any of the standard Java types.
B) It allows you to maintain a reference to an object that the garbage collector ignores.
C) It allows you to maintain a reference to an object in a different JVM, or on a remote machine.
D) It allows you to assign the reference to objects of different types.
E) It prevents the object from being saved when your class is serialized.

Sample Code

```
public class SwingApplet extends JApplet{
public void init(){
```

```
JList list = new JList();
list.addElement("Hello");
list.addElement("World");
getContentPane().add(list);
}
}
```

Given the sample applet code above, what would be displayed when the code is run?

A) The applet will be displayed with a JList component containing two elements, occupying the entire screen area of the applet.
B) The applet will be displayed with a JList component containing two elements, occupying the top-left corner of the applet.
C) Nothing - the code would not compile because you cannot add data directly to a JList. You must create a DefaultListModel to hold the data.
D) Nothing - the code would not compile because JApplet does not have a getContentPane() method. You should add the list directly to the JApplet.
E) The code will compile, but nothing will be displayed because you must override the paint() method of JApplet and call the list's paint() method.

Which class CANNOT be directly instantiated?

A) java.io.StringBufferInputStream
B) java.io.FileReader
C) java.io.DataInputStream
D) java.io.BufferedReader
E) java.io.InputStream

Which of the following is true for untrusted applets running on most browsers?

A) Applets can read to or write from files in the "/tmp" or "c:\tmp" directory.
B) Applets can change the ClassLoader.
C) Applet frames display a warning.
D) Applets can connect to computers other than the codebase.
E) Applets can retrieve the user's account name.

Sample Code

```
class A {
B b = new B();
C c = (C) b;
}
```

When will the cast of "b" to class C be allowed in the code above?

A) B and C are subclasses of the same superclass.
B) C is a final class.
C) If B and C are superclasses of the same subclass.
D) B is a superclass of C.
E) B is a subclass of C.

When could you implement the java.io.Externalizable interface?

A) It will allow other virtual machines to retrieve an object.
B) To create a JavaBean used in visual programming environments.
C) It will make a class's public methods accessible outside the virtual machine.
D) When you must store an object on a different platform.
E) When you need control over the information recorded when placing objects in persistent storage.

If two objects with non-"transient" references to each other are serialized using ObjectOutputStream, will this create an infinite loop? Why or why not?

A) Yes, since the references are not "transient".
B) No, ObjectOutputStream ensures that each object is written only once.
C) Yes, ObjectOutputStream does not keep track of previously stored objects.
D) No, ObjectOutputStream only stores relative memory references.
E) No, ObjectOutputStream does not store referenced objects.

Which Java construct maps to the CORBA IDL 'module' construct?

A) method
B) package
C) class
D) interface
E) static variable

What code would you use to retrieve an image from a remote website in a Java application (i.e., NOT an applet)?

A) URL url = new URL("http://www.images.com/test.jpg"); Image img = url.openConnection().getContent();
B) URL url = new URL("http://www.images.com/test.jpg"); Image img = Toolkit.getDefaultToolkit().getImage(url);
C) URL url = new URL("http://www.images.com/test.jpg"); Image img = (new AppletContext()).getImage(url);
D) URL url = new URL("http://www.images.com/test.jpg"); Image img = Runtime.getRuntime().load(url);
E) URL url = new URL("http://www.images.com/test.jpg"); Image img = (new Applet()).getImage(url);

Sample Code

```
import java.sql.*;
Additional code here
Connection con = DriverManager.getConnection(url);
String s = "call proc(?,?,?,?)";
CallableStatement cs = con.prepareCall(s);
cs.setInt(1,1);
cs.setInt(2,2);
cs.setInt(3,3);
cs.registerOutParameter(4,java.sql.TYPES.INTEGER);
```

In reference to the above, which command will execute procedure proc(), which includes two INSERT sql statements and one OUT parameter?

A) con.getConnection(cs)
B) ResultSet = cs.runQuery()
C) cs.executeQuery()
D) cs.execute()
E) cs.prepareCall().execute()

What class will open a TCP connection to a server for data communications using a proprietary application protocol?

A) java.net.ServerSocket
B) java.net.Socket
C) java.net.SocketImpl
D) java.net.DatagramSocket
E) java.net.URLConnection

What is NOT a typical feature of visual JavaBeans?

A) Properties, both for customization and for programmatic use.
B) Customization, so that when using an application builder a user can customize the appearance and behavior of a bean.
C) Events, so that a simple communication metaphor can be used to connect up beans.
D) Distributed framework, so the visual component resides on the client, while the logic resides on the server.
E) Persistence, so that a bean can be customized in an application builder and then have its customized state saved away and reloaded later.

In JavaIDL programming, the NamingContext allows you to bind objects to the naming service and look up objects. Which code fragment correctly gets a reference to the NamingContext (given that 'orb' is a reference to the ORB)?

A) org.omg.CORBA.Any any = orb.resolve_name_service("NameService");

NamingContext nc = NamingContextHolder.convert(any);
B) NamingContext nc = orb.getNamingContext("NameService");
C) java.lang.Object o = orb.lookup("NameService"); NamingContext nc = (NamingContext) o;
D) org.omg.CORBA.Object o = orb.resolve_initial_references("NameService");
NamingContext nc = NamingContextHelper.narrow(o);
E) org.omg.CORBA.ObjectHolder oh = orb.lookup_reference("NameService");
NamingContext nc = ObjectHelper.toNamingContext(oh);

How would you add data to a Swing component such as a JTable, JTree, or JList?

A) Add the data to the component's model.
B) Add the data directly to the component.
C) Add the data to the component's view.
D) Add the data to the component's peer.
E) Add the data to the component's controller.

Which Java API would you use while writing Java objects that act as servers for distributed CORBA objects written in C++?

A) JDBC
B) RMI
C) Infobus
D) JavaIDL
E) JNI

Which of the following is NOT a reserved java keyword?

A) boolean
B) final
C) finally
D) safe
E) native

What is the purpose of the java.text package?

A) To allow formatting of text data for specific geographic locations.
B) To render text strings on java.awt.Container objects.
C) To facilitate spelling and grammar checks.
D) To perform font management between different platforms.
E) To provide classes for manipulating Strings.

When an object is deserialized, how does the interpreter determine that the version of the class file matches the serialized data?

A) A unique identifier is included in the serialization stream that can be used to determine class definition compatibility.
B) The serialized object stores the proper class name.
C) The serialized object stores a URL to the proper class.
D) The ClassLoader takes care of testing for deserialization.
E) The ClassLoader compares the hashCodes.

Sample Code

```
class A{ }
class B extends A{ }
class C extends A{ }
class Test{
public static void main(String[] args){
A a = new A();
B b = new B();
C c = new C();
// assignment here
}
}
```

Given the above sample code, which of the following assignments would be legal?

A) None of the above.

B) c=a

C) b=c

D) a=b

E) b=a

When are constructors invoked?

A) When the object needs garbage collection.

B) When any method is invoked on an object.

C) When the java virtual machine begins garbage collection.

D) When a new instance of a class is instantiated.

E) When a superclass object is instantiated.

Which Java API would you use to look up an Enterprise JavaBean (EJB) component?

A) JDBC

B) Jini

C) JNDI

D) JMAPI

E) JMS

Sample Code

```
public class Applet2D extends JApplet{
public void paint(Graphics g){
Graphics2D g2d = // *** insert code here ***
// further graphics code ...
}
}
```

Given the above sample code, how would you obtain a reference to a Graphics2D object to allow you to use the advanced graphics functions supported by the Java 2D API?

A) Graphics2D g2d = SwingUtilities.getGraphics2D();

B) Graphics2D g2d = UIManager.getGraphics2D();

C) Graphics2D g2d = g.getGraphics2D();

D) Graphics2D g2d = (Graphics2D) g;

E) Graphics2D g2d = new Graphics2D(g);

Sample Code

```
public void parseUserList( List users ){
Iterator iterator = users.iterator();
while( iterator.hasNext() ){
String user = iterator.next();
if( user.equals("Smith") )
iterator.remove();
}
}
```

The above sample code is a method that takes a list of usernames and parses them to remove any users named "Smith". What is wrong with the code?

A) Iterator does not have a method called next(); it should be nextElement().

B) Iterator does not have a method called hasNext(); it should be hasMoreElements().

C) Iterator does not support removing elements from the underlying List; you should use users.remove("Smith").

D) next() returns an Object, NOT a String, so you have to cast the returned value to a String.

E) List does not have a method called iterator(); it should be getIterator().

Sample Code

```
public void printIt(String txt) {
StringTokenizer st = new StringTokenizer(txt);
while (st.hasMoreTokens()) {
System.out.println(st.nextToken());
}
}
```

Referring to the above, what is the result when the following statement is invoked?
printIt(" Hello\n World\t!");

A) The following is outputted:

```
HelloWorld!
```

B) The following is outputted:

```
Hello
World!
```

C) The following is outputted:

```
Hello
World
!
```

D) java.util.NoSuchElementException is thrown.

E) The following is outputted:

```
Hello
World
!
```

Sample Code

```
package B;
public class A {
int getSquare(int i) {
return i*i;
}
}
```

Referring to the above, what objects can access method getSquare() in class A?

A) Class A, all subclasses of A, classes in package B, and all classes in sub-packages of B.

B) Class A and its subclasses only.

C) Class A, all subclasses of A, and classes in package B only.

D) Class A and classes in package B only.

E) Class A.

Sample Code

```
class A {
Object get(Object o) {
try {
return o.getClass().getName();
} catch (Exception e) {
return null;
}
}
}
```

Referring to the above, what is the expected output when the following code is executed?

```
A a = new A();
Vector v = new Vector();
System.out.println( a.get( v ) );
```

A) A

B) null

C) java.util.Vector

D) NullPointerException

E) Object

Sample Code

```
String readInput() {
try {
```

```
char buf[] = new char[80];
Reader in = new BufferedReader(System.in);
in.read(buf, 0, 80);
return new String(buf);
} catch (Exception e ) {}
return "";
}
```

What is wrong with method readInput() above?

A) String objects cannot be instantiated with a character array.

B) java.io.IOException must be caught instead of java.lang.Exception.

C) BufferedReader cannot convert standard input into a Reader.

D) Standard input cannot be used with other input streams.

E) InputStream must be used with an InputStream.

In a CORBA appSample Code

```
public void print(Printable p){
PrinterJob pj = PrinterJob.getPrinterJob();
PageFormat pf = pj.pageDialog(pj.defaultPage());
pj.setPrintable(p,pf);
// now get user to select properties and confirm print
}
```

Given the above sample code, which takes a Printable object and sets up a PrinterJob to print that object, what code would you use to allow the user to select properties for the print job, confirm, and then print?

A) pj.printerDialog(pj.getDefaultPrinterOptions()); if(pj.printConfirmed()) p.print();

B) p.print();

C) Properties props = pj.getUserOptions();
if(((Boolean)props.getValue(PrinterJob.CONFIRM)).equals(true)) p.print(pj);

D) PrinterDialog pd = new PrinterDialog(); if(pd.show() == Dialog.OK) pj.print(p);

E) if(pj.printDialog()) pj.print();

lication, how would a client create a new remote object on the server?

A) By using a naming service, such as the tnameserv in JDK 1.2, to create the object.

B) By using com.omg.CORBA.ORB.string_to_object() method to instantiate a new object from a stringified object reference.

C) By using the server objects Home interface which provides methods for finding, creating, and deleting objects.

D) By using the Remote Object Activation facility.

E) By calling an existing server object's "factory" method which would create a new object and return a reference to the client.

What are the two main types of persistence for entity EJBs?

A) Application-managed and Database-managed

B) Applet-managed and Servlet-managed

C) Client-managed and Server-managed

D) Context-managed and Session-managed

E) Bean-managed and Container-managed

```
String username = Runtime.getParameter("username");
String us2ername = ar[0];
String u3sername = Class.getField("username");
String username4 = System.getenv("username");
String username5 = System.getProperty("username");
```

Sample Code

```
class A {
B b = new B();
C c = (C) b;
}
```

When will the cast of "b" to class C be allowed in the code above?
A) C is a final class.
B) B is a subclass of C.
C) B and C are subclasses of the same superclass.
D) B is a superclass of C.
E) If B and C are superclasses of the same subclass.
Sample Code

```
package mypackage;
class MyException extends java.lang.Exception {
public MyException() { super(); }
public MyException( String s ) { super(s); }
}
```

Referring to the above, which code segment properly generates an exception of type "MyException"?
A) MyException e = new MyException();
return e;
B) new Exception(MyException);
C) catch(MyException e) { }
D) exception new MyException();
E) throw new MyException("Error Occurred!");
Sample Code

```
class A {
int i, j, k;
public A(int ii) { i = ii; }
public A() {
k = 1;
}
}
```

What code will instantiate an object of class A in the code above?
A) A a = new A(3.3);
B) A a = new A(4,8);
C) A a = new A(3);
D) A(3) a;
E) new A(this);
Which interface is intended to supercede Enumeration in the collections framework?
A) Iterator
B) List
C) Vector
D) Set
E) Comparator
Which of the following services is NOT provided by Java RMI?
A) Naming service
B) Distributed garbage collection
C) Distributed transaction management
D) Dynamic class loading
E) Remote object activation
How would you create a scrollable ResultSet?
A) Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATEABLE); ResultSet rs = st.executeQuery(sqlString);
B) ResultSet rs = new ResultSet(); rs.setScrollable(true); rs.execute(sqlString);
C) Statement st = new Statement(); st.setCursorType(Statement.TYPE_SCROLLABLE, Statement.CONCUR_READ_ONLY);
ResultSet rs = st.executeSQL(sqlString);
D) ScrollableResultSet srs = new ScrollableResultSet();

srs.openResultSet(sqlString);

E) Statement st = con.createStatement(); ResultSet rs = st.executeQuery(sqlString, ResultSet.SCROLLABLE);

Sample Code

```
class A implements Runnable {
```

What code will create a daemon thread of a class A object with the code above?

A) Thread t = new Thread(A, true);

B) Daemon t = new Thread(new A());

C) A a = new A().setDaemon(true);
Thread t = new Thread(a);

D) A a = new A();
Thread t = new Thread(a);
t.setDaemon(true);

E) A a = new A();
Daemon t = new Thread(a);

How can you know when a user closes a window so you can perform cleanup?

A) Java does cleanup automatically when a window is closed.

B) By implementing FrameListener and placing cleanup code in a frameClosing(FrameEvent e) method.

C) By creating an inner class called WindowListener, extending WindowEventHandler, then putting cleanup code in a processEvent(Event e) method.

D) By placing cleanup code in a try - catch block that catches a FrameClosingException.

E) By implementing WindowListener and putting cleanup code in a public void windowClosing(WindowEvent e) method.

Sample Code

```
String sql = "SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEE WHERE ID='123'";
Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATEABLE);
ResultSet rs = st.executeQuery(sql);
rs.first();
rs.updateString(1,"Tom");
rs.updateString(2,"Smith");
```

Given the above sample code, what would you insert after the last line in order to write the changes to the database?

A) rs.close();

B) con.commit();

C) st.saveChanges();

D) rs.updateRow();

E) rs.next();

Sample Code

```
public void createTempFiles(String d) {
File f = new File(d);
f.mkdirs();
// more code here ...
}
```

What is the result when the following statement is invoked on the code above on a Unix platform?

createTempFiles("/tmp/myfiles/_3214");

A) The "myfiles" directory is created in the "/tmp" directory.

B) A java.lang.SecurityException is thrown.

C) The directory "/tmp/myfiles/_3214" is created if it doesn't already exist.

D) The file "_3214" is created in the "/tmp/myfiles" directory.

E) A java.io.DirectoryNotCreatedException is thrown.

Sample Code

```
public class TestServlet extends HttpServlet{
public void doGet(HttpServletRequest request, HttpServletResponse response){
// insert code here
}
}
```

Given the above sample servlet code, what code fragment could you insert to print out the name and value of the cookies passed from the client that initiated this reqest?

A) Cookie[] cookies = request.getCookies(); for(int i=0; i<cookies.length; i++) System.out.println(cookies[i].getName()+","+cookies[i].getValue());

B) Enumeration cookies = response.getCookies(); while(cookies.hasMoreElements()){ Cookie cookie = (Cookie)cookies.nextElement(); System.out.println(cookie.getName()+","+cookie.getValue()); }

C) The Servlet API does NOT support the use of cookies, since they may not be available in all browsers.

D) String[] names = request.getSession().getValueNames(); for(int i=0; i<names.length; i++){ String cookieValue = request.getSession().getValue(names[i]); System.out.println(names[i]+","+cookieValue); }

E) The Servlet API only supports the use of cookies for maintaining a sessionID on the client.

Sample Code

```
public void paint(Graphics g){
Graphics2D g2d = (Graphics2D) g;
// turn off antialiasing here
// display text
}
```

Given the above sample code, how would you turn off text antialiasing before you display the text?

A) g2d.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING, RenderingHints.VALUE_TEXT_ANTIALIAS_OFF);

B) RenderingHints hint = new RenderingHints(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_OFF); g2d.addRenderingHint(rh);

C) SwingUtilities.setAntialisingEnabled(false);

D) UIManager.setRenderingHints(Graphics2D.TEX_ANTIALIASING_ENABLED, false);

E) You CANNOT control whether antialiasing is enabled. It depends on the native windowing system.

Sample Code

```
public class SaveObject{
public static void main( String[] args ){
Employee smith = new Employee("012345","Smith","James","Payroll");
FileOutputStream fos = new FileOutputStream("smith.dat");
ObjectOutputStream oos = new ObjectOutputStream(fos);
oos.writeObject(smith);
fos.close();
}
}
```

Given the above sample code, which of the following would prevent the code from working properly?

A) The class Employee does NOT implement Serializable.

B) The file "smith.dat" does NOT already exist.

C) The class Employee implements Externalizable.
D) The file "smith.dat" already exists.
E) The class Employee is declared as transient.

In JavaIDL programming, the NamingContext allows you to bind objects to the naming service and look up objects. Which code fragment correctly gets a reference to the NamingContext (given that 'orb' is a reference to the ORB)?

A) NamingContext nc = orb.getNamingContext("NameService");
B) org.omg.CORBA.ObjectHolder oh = orb.lookup_reference("NameService"); NamingContext nc = ObjectHelper.toNamingContext(oh);
C) org.omg.CORBA.Any any = orb.resolve_name_service("NameService"); NamingContext nc = NamingContextHolder.convert(any);
D) java.lang.Object o = orb.lookup("NameService"); NamingContext nc = (NamingContext) o;
E) org.omg.CORBA.Object o = orb.resolve_initial_references("NameService"); NamingContext nc = NamingContextHelper.narrow(o);

Which of the following types of audio files does JDK 1.2 NOT support?

A) WAV
B) RMF
C) MIDI
D) AU
E) MP3

Sample Code

```
public class ArrayTest{
public static void main(String[] args){
// insert code here
}
}
```

Given the above sample code, what would you insert to print out a list of the command-line arguments?

A) for(int i=0; i<args.length(); i++) System.out.println(args[i]);
B) for(int i=0; i<sizeof(args); i++) System.out.println(args[i]);
C) for(int i=0; i<args.length; i++) System.out.println(args[i]);
D) for(int i=0; i<args.getLength(); i++) System.out.println(args[i]);
E) for(int i=0; i<args.size(); i++) System.out.println(args[i]);

When designing a custom GUI component, which interface should you implement to provide support for assistive technologies such as screen magnifiers and speech recognition?

A) You do NOT have to implement an interface; simply have your component extend JComponent to inherit the functionality.
B) Observer
C) Externalizable
D) Accessible
E) InputMethodListener

Sample Code

```
static {
Additional code here
}
```

In reference to the above, what does a static code block mean?

A) Internal variables are erased when the object is written to persistent storage.
B) All methods and variables defined inside block are implicitly static.
C) All internal variables must be "static".
D) The internal code will be executed once when the class is first loaded.
E) A compiler error will occur due to an incomplete method definition.

Sample Code

```
class Employee{
```

```
protected String name, salary;
public Employee() {}
public Employee(String aName, String aSalary) {
name=aName; salary=aSalary;
}
public String toString(){ return (name + ", " + salary); }
}
class Manager extends Employee{
protected String car;
public Manager(String aName, String aSalary, String aCar) {
car = aCar;
super( aName, aSalary );
}
public String toString(){ return (name + ", " + salary + ", " + car); }
}
public class EmployeeTest{
public static void main( String[] args ){
Employee[] employees = new Employee[3];
employees[0] = new Employee("Smith","$25,000");
employees[1] = new Employee("Jones","$35,000");
employees[2] = new Manager("Wilson","$45,000","BMW");
for(int i=0; i<employees.length; i++)
System.out.println(employees[i]);
}
}
```

Given the above sample code, what will be the result when the code is run?

A) The code will NOT compile, because the Manager class cannot access name and salary from the Employee class.

B) Smith, $25,000 Jones, $35,000 Wilson, $45,000, BMW

C) The code will NOT compile, because System.out.println() does not take an Employee parameter.

D) The code will NOT compile, because "super(aName,aSalary);" must be the first line in the Manager constructor.

E) Smith, $25,000 Jones, $35,000 Wilson, $45,000

When are constructors invoked?

A) When the object needs garbage collection.

B) When the java virtual machine begins garbage collection.

C) When a superclass object is instantiated.

D) When any method is invoked on an object.

E) When a new instance of a class is instantiated.

Sample Code

```
class A {
int getAbs(int i) {
return Math.abs(i);
}
}
```

Why is an object of class Math was not instantiated before calling abs()?

A) Any class's methods can be invoked without instantiating first.

B) Math is a final class.

C) There is an error.

D) Because method abs() is static.

E) Math belongs to the java.lang package.

Sample Code

```
class MyXML implements java.io.Serializable {
transient String name;
transient String value;
Vector subTree = new Vector();
public MyXML() {}
public void insert(MyXML x) { subTree.addElement( x ); }
static final long serialVersionUID = 3487495895819393L;
}
```

What would happen if you modified the class above by removing the transient keywords and then loading a previously serialized object?

A) The object would be deserialized with name and value set to null.
B) A new "empty" instance of class MyXML would be created.
C) The object would load normally with name and value set to their previous state.
D) Nothing. A compiler error would occur because serialVersionUID is defined.
E) An exception would be thrown.

What is Java serialization?

A) Distributed persistence.
B) The ability transform the state of an object into bits and resurrect a copy of the object from those bits.
C) Remote method invocation.
D) Marshaling and unmarshaling of remote objects.
E) The ability to examine the encapsulated data of a class.

How can you ensure compatibility of persistent object formats between different class versions?

A) Add "transient" variables.
B) Implement java.io.Serilizable, add a serial version UID, and write your own readObject() method to handle the different versions.
C) Make the whole class transient.
D) Implement java.io.Serializable.
E) Override the default ObjectInputStream and ObjectOutputStream.

What are the two main types of persistence for entity EJBs?

A) Bean-managed and Container-managed
B) Client-managed and Server-managed
C) Context-managed and Session-managed
D) Applet-managed and Servlet-managed
E) Application-managed and Database-managed

What are the two main types of persistence for entity EJBs?

A) Bean-managed and Container-managed
B) Client-managed and Server-managed
C) Context-managed and Session-managed
D) Applet-managed and Servlet-managed
E) Application-managed and Database-managed

Sample Code

```
class A{ }
class B extends A{ }
class C extends A{ }
class Test{
public static void main(String[] args){
A a = new A();
B b = new B();
C c = new C();
// assignment here
}
}
```

Given the above sample code, which of the following assignments would be legal?

A) c=a
B) a=b

C) b=a

D) b=c

E) None of the above.

Sample Code

```
// C++ Constant Definition
#define ANSWER_TO_ULTIMATE_QUESTION 42
```

Referring to the code above, how would do you define an equivalent constant in Java?

A) static int ANSWER_TO_ULTIMATE_QUESTION = 42;

B) public transient int ANSWER_TO_ULTIMATE_QUESTION = 42;

C) public static native long ANSWER_TO_ULTIMATE_QUESTION = 42L;

D) static volatile int ANSWER_TO_ULTIMATE_QUESTION = 42;

E) public static final int ANSWER_TO_ULTIMATE_QUESTION = 42;

Sample Code

```
class HelloWorld extends java.awt.Canvas {
String str = "Hello World";
public void paint(java.awt.Graphics g) {
FontMetrics fm = g.getFontMetrics();
g.setColor(java.awt.Color.black);
Dimension d = getSize();
__?__
}
}
```

Which code segment below can be inserted into the method above to paint "Hello World" centered within the canvas?

A) g.drawText(str, d.width, d.height, Font.CENTERED);

B) g.drawString(str, d.width/2, d.height/2);

C) g.drawString(str, d.width/2 - fm.stringWidth(str)/2, d.height/2 - fm.getHeight()/2);

D) g.drawText(str, 0, 0);

E) g.translate(d.width/2 - fm.stringWidth(str), d.height/2 - fm.getHeight());
g.drawString(str, 0, 0);

Which of the following Swing components is NOT a lightweight component?

A) JButton

B) JPanel

C) JFrame

D) JRootPane

E) JList

Sample Code

```
class StopWatch {
java.util.Calendar cal1,cal2;
public void start() {
cal1 = java.util.Calendar.getInstance();
}
public void stop() {
cal2 = java.util.Calendar.getInstance();
}
public long getDifference() {
long diff = 0;
try {
diff = ________________________;
} catch (Exception e) {
diff = -1;
}
return diff;
}
}
```

Referring to the above, what should the local variable "diff" be set to, in the getDifference() method, so that the number of seconds elapsed is returned?

A) cal2.getTimeInMillis() - cal1.getTimeInMillis()

B) (long) (cal2 - cal1)

C) cal2.get(java.util.Calendar.SECOND) - cal1.get(java.util.Calendar.SECOND)

D) (cal2.getTime().getTime() - cal1.getTime().getTime()) / 1000

E) None, as the calendar class doesn't keep track of time, just dates.

How may an applet be sent unsolicited messages from a remote (server) application?

A) By installing a SecurityManager that permits listening on a port.

B) By creating and exporting a remote object, then sending a reference to the server for RMI updates.

C) By registering with a remote application on a platform different from that which served the applet.

D) By setting up a socket that listens on a specific port.

E) By setting the java.rmi.server.codebase property to the remote server location.

An applet requires a method in java.awt.Panel that became available in the java 1.1 release. If the applet loads in a browser that only supports java 1.0, what will the browser do?

A) It will request the class file from the web server.

B) It will throw a NoSuchMethodException, ignore the missing methods, and continue.

C) It will reject the entire web page and revert to the previous page.

D) It will open a window asking permission to load a core java class from an untrusted source.

E) It will throw a NoSuchMethodError and not load the applet.

Sample Code

```
public class Test{
public static void main(String[] args){
int x=5, y=7;
swap(x,y);
System.out.println("x = " + x + ", y = " + y );
}
static void swap(int a, int b){
int c = a;
a = b;
b = c;
}
}
```

Given the above sample code, what will be printed when the Test program is run?

A) x = 5, y = 7

B) x = 12, y = 12

C) x = 7, y = 5

D) x = 0, y = 0

E) The code will not compile, because the swap() method is not visible from main().

Sample Code

```
static {
System.loadLibrary("mNativeLib");
}
public int myMethod(int count) {
Additional code here
}
```

In reference to the above, if myMethod() is implemented in native code library "mNativeLib", how must its declaration be changed?

A) Replace it with native myMethod(int count) {}.

B) Move it into the static code block.

C) Replace it with public System.nativeMethod("myMethod(int)");.

D) Replace myMethod()'s code with System.execute("mNativeLib", myMethod(count));.

E) Replace it with public native int myMethod(int count);.

Widget

The widget shown above is similar to which standard java.awt class?

A) TextBox

B) Entry

C) TextField

D) TextArea

E) Label

Sample Code

```
public class RefTest{
private WeakReference wr;
public static void main(String[] args){
setWeak();
System.out.println("wr = " + wr.get());
}
public static void setWeak(){
String s = "Test";
wr = new WeakReference(s);
}
}
```

Given the above sample code, what will be printed when the code is run?

A) wr = null

B) A NullPointerException will be thrown.

C) wr = Test

D) It depends on whether the garbage collector has run. If it has NOT: wr = Test If it has: wr = null

E) The code will NOT compile, because the private variable wr is not visible to main().

What does it mean if a method is final synchronized?

A) Methods which are synchronized cannot be final.

B) Only one synchronized method can be invoked at a time for the entire class.

C) All final variables referenced in the method can be modified by only one thread at a time.

D) The method cannot be overridden and is therefore certain to be threadsafe.

E) This is the same as declaring the method private.

Sample Code

```
public void parseUserList( List users ){
Iterator iterator = users.iterator();
while( iterator.hasNext() ){
String user = iterator.next();
if( user.equals("Smith") )
iterator.remove();
}
}
```

The above sample code is a method that takes a list of usernames and parses them to remove any users named "Smith". What is wrong with the code?

A) Iterator does not have a method called next(); it should be nextElement().

B) Iterator does not have a method called hasNext(); it should be hasMoreElements().

C) next() returns an Object, NOT a String, so you have to cast the returned value to a String.

D) Iterator does not support removing elements from the underlying List; you should use users.remove("Smith").

E) List does not have a method called iterator(); it should be getIterator().

Sample Code

```
Object open(String clsName) throws Exception {
Class cls = Class.forName( clsName );
return cls.newInstance();
}
```

What would happen if the open() method as shown above was called with "java.io.FileInputStream" as the parameter?

A) A java.lang.InstantiationException exception is thrown.

B) A new java.io.FileStream instance is returned.

C) A java.io.FileNotFound exception is thrown.

D) Nothing. A compilation error would occur.

E) A java.lang.SecurityException exception is thrown.

Sample Code

```
import java.awt.*;
class A extends Frame implements ActionListener,
WindowListener {
Button button1;
public A() {
button1 = new Button("Click Me");
button1.addActionListener(this);
add(button1);
show();
}
public actionPerformed(ActionEvent e) {}
void b1() { return; }
}
```

In reference to the above, what statement should be added to actionPerformed() so that b1() is invoked for a click on button1?

A) if(e.getActionCommand().equals("button1")) b1();

B) if(e.type == ButtonAction) b1();

C) if(e.getSource().equals(button1)) b1();

D) if(e.target == button1) b1();

E) b1();

Sample Code

```
import java.awt.*;
class A extends Frame implements ActionListener,
WindowListener {
Button button1;
public A() {
button1 = new Button("Click Me");
button1.addActionListener(this);
add(button1);
show();
}
public actionPerformed(ActionEvent e) {}
void b1() { return; }
}
```

In reference to the above, what statement should be added to actionPerformed() so that b1() is invoked for a click on button1?

A) if(e.getActionCommand().equals("button1")) b1();

B) if(e.type == ButtonAction) b1();

C) if(e.getSource().equals(button1)) b1();
D) if(e.target == button1) b1();
E) b1();

Which web technology do Servlets replace?

A) CGI scripts
B) Cookies
C) Dynamic HTML
D) JavaScript
E) HTTPS

Sample Code

```
public class SetTest{
public static void main(String[] args){
SortedSet ss = new TreeSet();
for(int j=0; j<args.length; j++){
ss.add(args[j]);
}
Iterator i = ss.iterator();
while( i.hasNext() ){
System.out.println(i.next());
}
}
}
```

Given the above sample code, what will be printed when the program is run with the following command-line? >java SetTest red blue red green

A) blue green red
B) red blue
C) red blue red green
D) red blue green
E) blue green red red

Sample Code

```
public class Test{
public static void main(String[] args){
StringBuffer[] messages = new StringBuffer[5];
messages[0].append("Hello world!");
System.out.println("First message is " + messages[0]);
}
}
```

What will be the output when the above sample code is run?

A) First message is null.
B) First message is Hello World!
C) A NullPointerException will be thrown.
D) An ArrayIndexOutOfBounds will be thrown.
E) The code would not compile.

Sample Code

```
big_loop: for(int i = 0; i < 3; i++) {
try {
for(int j = 0; j < 3; j++) {
if (i==j) continue;
else if (i>j) continue big_loop;
System.out.print("A ");
}
}
finally {
System.out.print("B ");
}
System.out.print("C ");
}
```

What is the output from the program above?

A) A B C A B C A B C
B) A A A B C A A A B C A A A B C
C) A A B C B B
D) A A B B C A C A
E) None. The program will enter into an infinite loop.

When a remote class is loaded, how does java (JVM) ensure the bytecode is safe?

A) The RMIClassLoader verifies the digital signature of the class.
B) The RMIClassLoader performs bytecode verification.
C) The JVM performs bytecode verification.
D) If the SecurityManager object permits the load, it is considered safe.
E) Remote object stubs use a separate memory space inside the JVM.

What class could be subclassed to package several locale-specific versions of a set of greetings used by an application?

A) java.text.RuleBasedCollator
B) java.text.CollationKey
C) java.util.Locale
D) java.util.TimeZone
E) java.util.ResourceBundle

Which of the following is NOT an advantage of using Swing?

A) Swing components will run in all commercially available browsers.
B) Lightweight components give a standard look and feel across platforms.
C) It provides greater functionality than standard AWT components.
D) Pluggable look and feel allows the end-user to easily change the appearance of an application.
E) It uses native OS widgets for more compliant look and feel.

Sample Code

```
class A {
int i=0;
public A() { i=8; }
public static void main(String args[]) {
A h = new A();
while (h.i <= 10) h.doIt();
}
public static void doIt() {
i++;
System.out.println("Hello");
}
}
```

What will the above program do?

A) It will print "Hello" once.
B) It will print "Hello" twice.
C) It will not run because the proper packages have not been imported.
D) It will not compile because doIt() cannot reference non-static variable i.
E) It will print "Hello" 11 times.

What happens if a parameter is passed to a remote object (using RMI) which does NOT implement Remote or Serializable?

A) A java.rmi.MarshallException is thrown.
B) A java.lang.NullPointerException is thrown.
C) It is passed by value.
D) It is passed by reference.
E) The code will not compile.

Sample Screen

Object 1 Here
Object 2 Here Object 3 Here Object 4 Here
Object 5 Here

To display five objects as shown above using all available screen space, what Layout Manager would be easiest?

A) BorderLayout
B) CardLayout
C) GridLayout
D) null
E) FlowLayout

How can you know when a user closes a window so you can perform cleanup?

A) By creating an inner class called WindowListener, extending WindowEventHandler, then putting cleanup code in a processEvent(Event e) method.
B) By implementing FrameListener and placing cleanup code in a frameClosing(FrameEvent e) method.
C) By placing cleanup code in a try - catch block that catches a FrameClosingException.
D) By implementing WindowListener and putting cleanup code in a public void windowClosing(WindowEvent e) method.
E) Java does cleanup automatically when a window is closed.

Sample Code

```
public drawThickLine(Graphics2D g, x1, y1, x2, y2){
// set line thickness here
g.drawLine(x1,y1,x2,y2);
}
```

Given the above sample code, what code would you insert at the indicated point to make the method draw a line that is ten pixels wide?

A) g.setPen(new DefaultPen(10));
B) g.setLineWidth(10);
C) g.setBrush(new Brush(10.0));
D) g.setStroke(new BasicStroke(10.0f));
E) Stroke s = new Stroke(); s.setWidth(10); g.setStroke(s);

Which code segment could execute the stored procedure "countRecs()" located in a database server?

A) PreparedStatement pstmt = connection.prepareStatement("countRecs()");
pstmt.execute();
B) Statement stmt = connection.createStatement();
stmt.executeStoredProcedure("countRecs()");
C) Statement stmt = connection.createStatement();
stmt.execute("COUNTRECS()");
D) StoredProcedureStatement spstmt =
connection.createStoredProcedure("countRecs()");
spstmt.executeQuery();
E) CallableStatement cs = con.prepareCall("{call COUNTRECS}");
cs.executeQuery();

JDK 1.2 includes Reference objects such as WeakReference and PhantomReference. How would you create your own type of Reference object?

A) Extend Reference.
B) Write methods to interact with the garbage collector.
C) You CANNOT create your own Reference type.
D) Implement Reference.
E) Create a class with a Reference attribute.

Which Java API would you use while writing Java objects that act as servers for distributed CORBA objects written in C++?

A) JNI
B) JDBC
C) Infobus
D) JavaIDL
E) RMI

Which of the following commands starts the Java IDL name service on port 1234?

A) nameserver -InitialPort 1234
B) tnameserv -ORBInitialPort 1234
C) cosnaming -COSInitialPort 1234
D) idlregistry -IDLInitialPort 1234
E) nameservice 1234

What is Java serialization?

A) Distributed persistence.
B) The ability to examine the encapsulated data of a class.
C) Remote method invocation.
D) The ability transform the state of an object into bits and resurrect a copy of the object from those bits.
E) Marshaling and unmarshaling of remote objects.

How would you create a menu item "Save" with a shortcut key of "Ctrl+S"?

A) JMenuItem save = new JMenuItem("Save"); save.addShortcutKey(new KeyStroke(KeyStroke.S | KeyStroke.CTRL_KEY));
B) JMenuItem save = new JMenuItem("Save"); save.setMnemonic("Ctrl+S");
C) You would have to override the KeyPressed event of the top level Frame, and handle the "Ctrl+S" to call the menu item's actionPerformed.
D) JMenuItem save = new JMenuItem("Save");
save.setAccelerator(KeyStroke.getKeyStroke(
KeyEvent.VK_S,Event.CTRL_MASK));
E) JMenuItem save = new JMenuItem("Save");
save.enableShortcut(KeyEvent.CTRL_KEY + KeyEvent.S);

```
public void print(Printable p){
PrinterJob pj = PrinterJob.getPrinterJob();
PageFormat pf = pj.pageDialog(pj.defaultPage());
pj.setPrintable(p,pf);
// now get user to select properties and confirm print
}
```

Given the above sample code, which takes a Printable object and sets up a PrinterJob to print that object, what code would you use to allow the user to select properties for the print job, confirm, and then print?

A) p.print();
B) pj.printerDialog(pj.getDefaultPrinterOptions()); if(pj.printConfirmed()) p.print();
C) PrinterDialog pd = new PrinterDialog(); if(pd.show() == Dialog.OK) pj.print(p);
D) Properties props = pj.getUserOptions();
if(((Boolean)props.getValue(PrinterJob.CONFIRM)).equals(true)) p.print(pj);
E) if(pj.printDialog()) pj.print();

What is an RMI "stub"?

A) It acts as a client-side proxy to receive RMI calls and pass them to the server.
B) It receives RMI calls from the server to enable applet callbacks.
C) It is used by the client to find remote objects on a server.
D) It is the interface implemented by both the client and server objects.
E) It enables applets to make RMI calls without violating browser security.

Which of the following types of audio files does JDK 1.2 NOT support?

A) MIDI
B) RMF
C) AU
D) MP3

E) WAV

```
1: Date myDate = new Date();
2: DateFormat dateFormat = DateFormat.getDateInstance();
3: String myString = dateFormat.format(myDate);
4: System.out.println( "Today is " + myString );
```

How can you change line 2 above so that the date is displayed in the same format used in China?

A) DateFormat dateFormat = DateFormat.getDateInstance();
dateFormat.setLocale(java.util.Locale.CHINA);
B) DateFormat dateFormat = DateFormat.getDateInstance(
DateFormat.SHORT, java.util.Locale.CHINA);
C) java.util.Locale locale = java.util.Locale.getLocale("China");
DateFormat dateFormat = DateFormat.getDateInstance(locale);
D) DateFormat.setLocale(java.util.Locale.CHINA);
DateFormat dateFormat = DateFormat.getDateInstance();
E) java.util.Locale locale = java.util.Locale.getLocale(java.util.Locale.CHINA);
DateFormat dateFormat = DateFormat.getDateInstance(locale);

How would you implement a text field for entering a password in a Login dialog using Swing components?

A) By using a JTextField and call setEchoChar('*').
B) By using a JPasswordField.
C) By using a JTextField and adding an event handler for the KeyPressed event to echo a '*' character.
D) By creating your own class which extends JTextField and overrides the keyDown() method to echo a '*' character.
E) use a JTextComponent and call
setTextDisplay(JTextComponent.PASSWORD).

Which aspect(s) of the bean is used to modify or retrieve the JavaBeans properties?

A) Accessor methods
B) Persistence mechanism
C) propertyManager() method (required in all beans)
D) Event adapters
E) Associated java.beans.PropertyChangeListener class methods

```
class A {
Object get(Object o) {
try {
return o.getClass().getName();
} catch (Exception e) {
return null;
}
}
}
```

Referring to the above, what is the expected output when the following code is executed?

```
A a = new A();
Vector v = new Vector();
System.out.println( a.get( v ) );
```

A) NullPointerException
B) A
C) Object
D) null
E) java.util.Vector

What does double-buffering in animations do?

A) It draws the next frame to an off-screen image object before displaying to eliminate flicker.

B) It prevents "hanging" using a MediaTracker object to ensure all frames are loaded before beginning.

C) It puts multiple frame in the same file and uses cropImage() to select the desired frame.

D) It uses a small image as wallpaper and a transparent image as the actual frame.

E) It draws each image to the screen twice to eliminate white spots.

```
class Class1 {
int total=0;
public static void main(String args[]) {
doIt();
}
void doIt() {
for(int i=0;i<5;i++) total += i;
System.out.println(total);
}
}
```

In reference to the above, how should the first line of method main() be changed to call doIt()?

A) Class1();

B) Class1 class1 = new Class1();

C) (new Class1()).doIt();

D) Class1().doIt();

E) No change is necessary.

In a CORBA application, how would a client create a new remote object on the server?

A) By calling an existing server object's "factory" method which would create a new object and return a reference to the client.

B) By using com.omg.CORBA.ORB.string_to_object() method to instantiate a new object from a stringified object reference.

C) By using a naming service, such as the tnameserv in JDK 1.2, to create the object.

D) By using the Remote Object Activation facility.

E) By using the server objects Home interface which provides methods for finding, creating, and deleting objects.

```
static {
System.loadLibrary("mNativeLib");
}
public int myMethod(int count) {
Additional code here
}
```

In reference to the above, if myMethod() is implemented in native code library "mNativeLib", how must its declaration be changed?

A) Replace myMethod()'s code with System.execute("mNativeLib", myMethod(count));.

B) Replace it with native myMethod(int count) {}.

C) Replace it with public System.nativeMethod("myMethod(int)");.

D) Move it into the static code block.

E) Replace it with public native int myMethod(int count);.

Which of the following is a requirement of a JavaBean?

A) Define a java.beans.PropertyEditor.

B) Use of the java.beans.PropertyChangeListener.

C) Provide a no-argument constructor.

D) Implement the java.beans.BeanInfo interface.

E) Extend java.beans.BeanDescriptor.

How would you disable tooltips in a Swing based Java application?

A) Tool tip support is determined by the underlying windowing system, and cannot be changed by a Java application.

B) UIManager.disableToolTips();

C) ToolTipManager.sharedInstance().setEnabled(false);

D) For each component in the application call: JComponent.setToolTipText(null);

E) SwingUtilities.setToolTipsEnabled(false);

Which of the following is NOT a valid java.lang.String declaration?

A) String myString = new String();

B) String cde = "cde";

C) String myString = new String(5);

D) char data[] = {'a','b','c'};
String str = new String(data);

E) String myString = new String("Hello");

Which of the following is true for untrusted applets running on most browsers?

A) Applets can connect to computers other than the codebase.

B) Applets can read to or write from files in the "/tmp" or "c:\tmp" directory.

C) Applets can change the ClassLoader.

D) Applets can retrieve the user's account name.

E) Applet frames display a warning.

In reference to the above, when you click on the button the applet changes background colors. Which method must it call to do this?

A) repaint()

B) shade()

C) draw()

D) setbgcolor()

E) restart()

In what way is it possible to circumvent applet security restrictions?

A) Install a different SecurityManager.

B) Attach a trusted digital signature to the downloaded applet JAR file.

C) Replace the default ClassLoader.

D) Connect to a ActiveX control located in the same web page.

E) There is no way to circumvent applet security restrictions.

```
1: class C extends Thread {
2: public void run() {
3: while(true) {
4: System.out.println("Hello World!");
5: try {
6: sleep(100);
7: } catch(Exception e) {}
8: }
9: }
10: public static void main(String[] s) {
11: C c = new C();
12: c.start();
13: }
14: }
```

What is the result of the program above?

A) A java.lang.StackOverflowException is thrown.

B) The CPU is used complete up by the infinite loop.

C) Nothing. Compiler error on line 6.

D) The system exits immediately with little or no output.

E) "Hello World!" is printed forever.

```
public class TestApplet extends JApplet{
public void init(){
```

```
JButton b1 = new JButton("one");
JButton b2 = new JButton("two");
JButton b3 = new JButton("three");
JButton b4 = new JButton("four");
JButton b5 = new JButton("five");
getContentPane().add(b1);
getContentPane().add(b2);
getContentPane().add(b3);
getContentPane().add(b4);
getContentPane().add(b5);
}
}
```

Given the above sample code of a JApplet, what would be displayed when the applet is run?

A) Only the button "five", occupying the entire area of the applet.

B) Five buttons, "one","two","three","four","five", in a column top to bottom.

C) Five buttons, "one","two","three","four","five", in a row left to right.

D) Only the button "one", occupying the entire area of the applet.

E) Five buttons, "one","two","three","four","five", randomly located within the applet.

```
package mypackage;
class MyException extends java.lang.Exception {
public MyException() { super(); }
public MyException( String s ) { super(s); }
}
```

Referring to the above, which code segment properly generates an exception of type "MyException"?

A) exception new MyException();

B) new Exception(MyException);

C) catch(MyException e) { }

D) throw new MyException("Error Occurred!");

E) MyException e = new MyException();
return e;