```
11. class Converter {
12. public static void main(String[] args) {
13. Integer i = args[0];
14. int j = 12;
15. System.out.println("It is " + (j==i) + " that j==i.");
16. }
17. }
```

What is the result when the programmer attempts to compile the code and run it with the command java Converter 12?

A. It is true that j==i.
B. It is false that j==i.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 13.

Answer: D

QUESTION: 196
Given

```
class Foo {
static void alpha() { /* more code here */ }
void beta() { /* more code here */ }
}
```

Which two statements are true? (Choose two.)

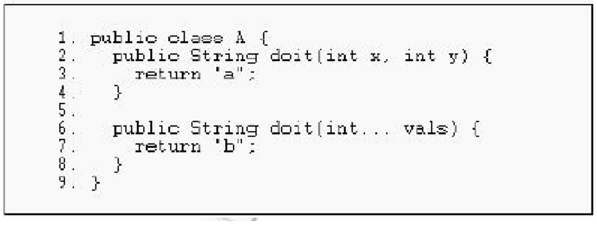
A. Foo.beta() is a valid invocation of beta().
B. Foo.alpha() is a valid invocation of alpha().
C. Method beta() can directly call method alpha().
D. Method alpha() can directly call method beta().

Answer: B,C

QUESTION: 197
Click the Exhibit button.
Given:

```
A a = new A();
System.out.println(a.doit(4, 5));
```

What is the result?



```
1. public class A {
2.   public String doit(int x, int y) {
3.     return "a";
4.   }
5.
6.   public String doit(int... vals) {
7.     return "b";
8.   }
9. }
```

A. Line 26 prints "a" to System.out.
B. Line 26 prints "b" to System.out.
C. An exception is thrown at line 26 at runtime.
D. Compilation of class A will fail due to an error in line 6.

Answer: A

QUESTION: 198
Which two code fragments correctly create and initialize a static array of int elements? (Choose two.)

A. static final int[] a = { 100,200 };
B. static final int[] a; static { a=new int[2]; a[0]=100; a[1]=200; }
C. static final int[] a = new int[2]{ 100,200 };
D. static final int[] a; static void init() { a = new int[3]; a[0]=100; a[1]=200; }

Answer: A,B

QUESTION: 199
Given:
5. class Atom {
6. Atom() { System.out.print("atom "); }
7. }
8. class Rock extends Atom {
9. Rock(String type) { System.out.print(type); }
10. }
11. public class Mountain extends Rock {
12. Mountain() {
13. super("granite ");
14. new Rock("granite ");
15. }
16. public static void main(String[] a) { new Mountain(); }
17. }
What is the result?

A. Compilation fails.
B. atom granite
C. granite granite
D. atom granite granite
E. An exception is thrown at runtime.
F. atom granite atom granite

Answer: F

QUESTION: 200
Given:

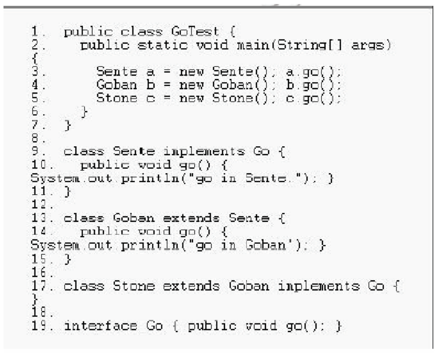
```
public class Plant {
private String name;
public Plant(String name) { this.name = name; }
public String getName() { return name; }
}
public class Tree extends Plant {
public void growFruit() { }
public void dropLeaves() { }
}
```

Which statement is true?

A. The code will compile without changes.
B. The code will compile if public Tree() { Plant(); } is added to the Tree class.
C. The code will compile if public Plant() { Tree(); } is added to the Plant class.
D. The code will compile if public Plant() { this("fern"); } is added to the Plant class.
E. The code will compile if public Plant() { Plant("fern"); } is added to the Plant class.

Answer: D

QUESTION: 201
Click the Exhibit button. What is the result?



```
public class GoTest {
  public static void main(String[] args)
{
    Sente a = new Sente(); a.go();
    Goban b = new Goban(); b.go();
    Stone c = new Stone(); c.go();
  }
}

class Sente implements Go {
  public void go() {
System.out.println("go in Sente."); }
}

class Goban extends Sente {
  public void go() {
System.out.println("go in Goban'); }
}

class Stone extends Goban implements Go {
}

interface Go { public void go(); }
```

A. go in Goban go in Sente go in Sente
B. go in Sente go in Sente go in Goban
C. go in Sente go in Goban go in Goban
D. go in Goban go in Goban go in Sente
E. Compilation fails because of an error in line 17.

Answer: C

QUESTION: 202
Which two classes correctly implement both the java.lang.Runnable and the java.lang. Cloneable interfaces? (Choose two.)

A. public class Session implements Runnable, Cloneable { public void run(); public Object clone(); }
B. public class Session extends Runnable, Cloneable { public void run() { /* do something */ } public Object clone() { /* make a copy */ }
C. public class Session implements Runnable, Cloneable { public void run() { /* do something */ } public Object clone() { /* make a copy */ }
D. public abstract class Session implements Runnable, Cloneable { public void run() { /* do something */ } public Object clone() { /*make a copy */ }
E. public class Session implements Runnable, implements Cloneable { public void run() { /* do something */ } public Object clone() { /* make a copy */ }

Answer: C,D

QUESTION: 203
Given:
11. class Mud {
12. // insert code here
13. System.out.println("hi");
14. }
15. }
And the following five fragments:
public static void main(String...a) {
public static void main(String.* a) {
public static void main(String... a) {
public static void main(String[]... a) {
public static void main(String...[] a) {
How many of the code fragments, inserted independently at line 12, compile?

A. 0

B. 1
C. 2
D. 3
E. 4
F. 5

Answer: D

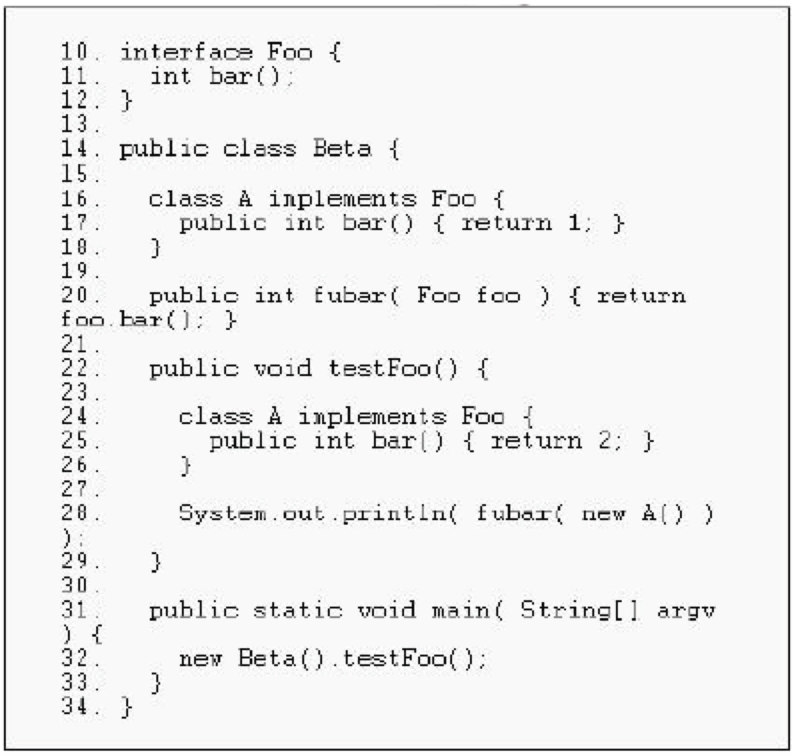
QUESTION: 204
Given:

```
11. public interface A111 {
12. String s = "yo";
13. public void method1();
14. }
17. interface B { }
20. interface C extends A111, B {
21. public void method1();
22. public void method1(int x);
23. } What is the result?
```

A. Compilation succeeds.
B. Compilation fails due to multiple errors.
C. Compilation fails due to an error only on line 20.
D. Compilation fails due to an error only on line 21.
E. Compilation fails due to an error only on line 22.
F. Compilation fails due to an error only on line 12.

Answer: A

QUESTION: 205
Click the Exhibit button.
Which three statements are true? (Choose three.)



```
10. interface Foo {
11.   int bar();
12. }
13.
14. public class Beta {
15.
16.   class A implements Foo {
17.     public int bar() { return 1; }
18.   }
19.
20.   public int fubar( Foo foo ) { return
foo.bar(); }
21.
22.   public void testFoo() {
23.
24.     class A implements Foo {
25.       public int bar() { return 2; }
26.     }
27.
28.     System.out.println( fubar( new A() )
);
29.   }
30.
31.   public static void main( String[] argv
) {
32.     new Beta().testFoo();
33.   }
34. }
```

A. Compilation fails.
B. The code compiles and the output is 2.
C. If lines 16, 17 and 18 were removed, compilation would fail.
D. If lines 24, 25 and 26 were removed, compilation would fail.
E. If lines 16, 17 and 18 were removed, the code would compile and the output would be 2.
F. If lines 24, 25 and 26 were removed, the code would compile and the output would be 1.

Answer: B,E,F

QUESTION: 206

Given:
11. class Alpha {
12. public void foo() { System.out.print("Afoo "); }
13. }
14. public class Beta extends Alpha {
15. public void foo() { System.out.print("Bfoo "); }
16. public static void main(String[] args) {
17. Alpha a = new Beta();
18. Beta b = (Beta)a;
19. a.foo(); 20. b.foo();

```
21. }
22. }
```

What is the result?

A. Afoo Afoo
B. Afoo Bfoo
C. Bfoo Afoo
D. Bfoo Bfoo
E. Compilation fails.
F. An exception is thrown at runtime.

Answer: D

QUESTION: 207

Given:

```
1. public class Threads4 {
2. public static void main (String[] args) {
3. new Threads4().go();
4. }
5. public void go() {
 6. Runnable r = new Runnable() {
7. public void run() {
8. System.out.print("foo");
9. }
10. };
11. Thread t = new Thread(r);
12. t.start();
13. t.start();
14. }
15. }
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. The code executes normally and prints "foo".
D. The code executes normally, but nothing is printed.

Answer: B

QUESTION: 208

Given:

```
1. public class TestOne {
2. public static void main (String[] args) throws Exception {
```

```
3. Thread.sleep(3000);
4. System.out.println("sleep");
5. }
6. }
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. The code executes normally and prints "sleep".
D. The code executes normally, but nothing is printed.

Answer: C

QUESTION: 209

Given:

```
 1. public class Threads3 implements Runnable {
2. public void run() {
3. System.out.print("running");
4. }
5. public static void main(String[] args) {
6. Thread t = new Thread(new Threads3());
7. t.run();
8. t.run();
9. t.start();
10. }
11. }
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. The code executes and prints "running".
D. The code executes and prints "runningrunning".
E. The code executes and prints "runningrunningrunning".

Answer: E

QUESTION: 210

Given:

```
public class NamedCounter {
private final String name;
private int count;
public NamedCounter(String name) { this.name = name; }
public String getName() { return name; }
public void increment() { count++; }
public int getCount() { return count; } public void reset() { count = 0;
}
```

Which three changes should be made to adapt this class to be used safely by multiple threads? (Choose three.)

A. declare reset() using the synchronized keyword
B. declare getName() using the synchronized keyword
C. declare getCount() using the synchronized keyword
D. declare the constructor using the synchronized keyword
E. declare increment() using the synchronized keyword

Answer: A,C,E

QUESTION: 211

Given that Triangle implements Runnable, and:

```
31. void go() throws Exception {
32. Thread t = new Thread(new Triangle());
33. t.start();
34. for(int x = 1; x < 100000; x++) {
35. //insert code here
36. if(x%100 == 0) System.out.print("g");
37. } }
38. public void run() {
39. try {
40. for(int x = 1; x < 100000; x++) {
41. // insert the same code here
42. if(x%100 == 0) System.out.print("t");
43. }
44. } catch (Exception e) { }
45. }
```

Which two statements, inserted independently at both lines 35 and 41, tend to allow both threads to temporarily pause and allow the other thread to execute? (Choose two.)

A. Thread.wait();
B. Thread.join();
C. Thread.yield();
D. Thread.sleep(1);
E. Thread.notify();

Answer: C,D

QUESTION: 212

Given:

```
public class TestSeven extends Thread {
private static int x;
public synchronized void doThings() {
int current = x;
current++;
x = current;
}
public void run() {
doThings();
}
}
```

Which statement is true?

A. Compilation fails.
B. An exception is thrown at runtime.
C. Synchronizing the run() method would make the class thread-safe.
D. The data in variable "x" are protected from concurrent access problems.
E. Declaring the doThings() method as static would make the class thread-safe.
F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.

Answer: E

QUESTION: 213

Given:

```
public class Yikes {

public static void go(Long n) {System.out.print("Long ");}
public static void go(Short n) {System.out.print("Short ");}
public static void go(int n) {System.out.print("int ");}
public static void main(String [] args) {
short y = 6;
long z = 7;
go(y);
go(z);
}
}
```

What is the result?

A. int Long
B. Short Long
C. Compilation fails.

D. An exception is thrown at runtime.

Answer: A

QUESTION: 214
Given:
12. Date date = new Date();
13. df.setLocale(Locale.ITALY);
14. String s = df.format(date);
The variable df is an object of type DateFormat that has been initialized in line 11. What is the result if this code is run on December 14, 2000?

A. The value of s is 14-dic-2000.
B. The value of s is Dec 14, 2000.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 13.

Answer: D

QUESTION: 215
Which two scenarios are NOT safe to replace a StringBuffer object with a StringBuilder object? (Choose two.)

A. When using versions of Java technology earlier than 5.0.
B. When sharing a StringBuffer among multiple threads.
C. When using the java.io class StringBufferInputStream.
D. When you plan to reuse the StringBuffer to build more than one string.

Answer: A,B

QUESTION: 216
Given that c is a reference to a valid java.io.Console object, and:
11. String pw = c.readPassword("%s", "pw: ");
12. System.out.println("got " + pw);
13. String name = c.readLine("%s", "name: ");
14. System.out.println(" got ", name);
If the user types fido when prompted for a password, and then responds bob when prompted for a name, what is the result?

A. pw: got fido name: bob got bob
B. pw: fido got fido name: bob got bob
C. pw: got fido name: bob got bob

D. pw: fido got fido name: bob got bob
E. Compilation fails.
F. An exception is thrown at runtime.

Answer: E

QUESTION: 217
Given:

```
String test = "This is a test";
String[] tokens = test.split("\s");
System.out.println(tokens.length);
```

What is the result?

A. 0
B. 1
C. 4
D. Compilation fails.
E. An exception is thrown at runtime.

Answer: D

QUESTION: 218
Given:

```
import java.io.*;
class Animal {
Animal() { System.out.print("a"); }
}
class Dog extends Animal implements Serializable {
Dog() { System.out.print("d"); }
}
public class Beagle extends Dog { }
```

If an instance of class Beagle is created, then Serialized, then deSerialized, what is the result?

A. ad
B. ada
C. add
D. adad
E. Compilation fails.
F. An exception is thrown at runtime.

Answer: B

QUESTION: 219

Given:

```
11. double input = 314159.26;
12. NumberFormat nf = NumberFormat.getInstance(Locale.ITALIAN);
13. String b;
14. //insert code here
```

Which code, inserted at line 14, sets the value of b to 314.159,26?

A. b = nf.parse(input);
B. b = nf.format(input);
C. b = nf.equals(input);
D. b = nf.parseObject(input);

Answer: B

QUESTION: 220

Given:

```
1. public class Target {
2. private int i = 0;
3. public int addOne(){
4. return ++i;
5. }
6. }
```

And:

```
1. public class Client {
2. public static void main(String[] args){
3. System.out.println(new Target().addOne());
4. }
5. }
```

Which change can you make to Target without affecting Client?

A. Line 4 of class Target can be changed to return i++;
B. Line 2 of class Target can be changed to private int i = 1;
C. Line 3 of class Target can be changed to private int addOne(){
D. Line 2 of class Target can be changed to private Integer i = 0;

Answer: D

QUESTION: 221

Given:

```
 11. class Animal { public String noise() { return "peep"; } }
12. class Dog extends Animal {
```

```
13. public String noise() { return "bark"; }
14. }
15. class Cat extends Animal {
16. public String noise() { return "meow"; }
17. }
...
30. Animal animal = new Dog();
31. Cat cat = (Cat)animal;
32. System.out.println(cat.noise());
```

What is the result?

A. peep
B. bark
C. meow
D. Compilation fails.
E. An exception is thrown at runtime.

Answer: E

QUESTION: 222

Given:

```
10. abstract class A {
11. abstract void a1();
12. void a2() { }
13. }
14. class B extends A {
15. void a1() { }
16. void a2() { }
17. }
18. class C extends B { void c1() { } } and: A x = new B(); C y = new C(); A z = new C();
```

What are four valid examples of polymorphic method calls? (Choose four.)

A. x.a2();
B. z.a2();
C. z.c1();
D. z.a1();
E. y.c1();
F. x.a1();

Answer: A,B,D,F

QUESTION: 223

Given:

```
3. class Employee {
4. String name; double baseSalary;
5. Employee(String name, double baseSalary) {
6. this.name = name;
7. this.baseSalary = baseSalary;
8. }
9. }
10. public class SalesPerson extends Employee {
11. double commission;
12. public SalesPerson(String name, double baseSalary, double commission) {
13. // insert code here
14. }
15. }
```

Which two code fragments, inserted independently at line 13, will compile? (Choose two.)

A. super(name, baseSalary);
B. this.commission = commission;
C. super(); this.commission = commission;
D. this.commission = commission; super();
E. super(name, baseSalary); this.commission = commission;
F. this.commission = commission; super(name, baseSalary);
G. super(name, baseSalary, commission);

Answer: A,E

QUESTION: 224

A team of programmers is involved in reviewing a proposed design for a new utility class. After some discussion, they realize that the current design allows other classes to access methods in the utility class that should be accessible only to methods within the utility class itself. What design issue has the team discovered?

A. Tight coupling
B. Low cohesion
C. High cohesion
D. Loose coupling
E. Weak encapsulation
F. Strong encapsulation

Answer: E

QUESTION: 225

Given that: Gadget has-a Sprocket and Gadget has-a Spring and Gadget is-a Widget and Widget has-a Sprocket Which two code fragments represent these relationships? (Choose two.)

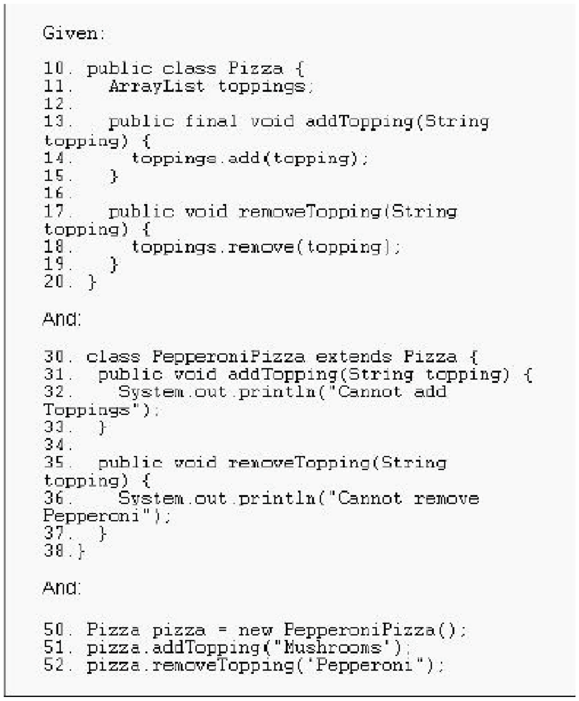
A. class Widget { Sprocket s; } class Gadget extends Widget { Spring s; }
B. class Widget { } class Gadget extends Widget { Spring s1; Sprocket s2; }
C. class Widget { Sprocket s1; Spring s2; } class Gadget extends Widget { }
D. class Gadget { Spring s; } class Widget extends Gadget{ Sprocket s; }
E. class Gadget { } class Widget extends Gadget{ Sprocket s1; Spring s2; }
F. class Gadget { Spring s1; Sprocket s2; } class Widget extends Gadget{ }

Answer: A,C

QUESTION: 226
Given:

```
1. class Pizza {
2. java.util.ArrayList toppings;
3. public final void addTopping(String topping) {
4. toppings.add(topping);
5. }
6. }
7. public class PepperoniPizza extends Pizza {
8. public void addTopping(String topping) {
9. System.out.println("Cannot add Toppings");
10. }
11. public static void main(String[] args) {
12. Pizza pizza = new PepperoniPizza();
13. pizza.addTopping("Mushrooms");
14. }
15. }
```

What is the result?



Given:

```
10. public class Pizza {
11.   ArrayList toppings;
12.
13.   public final void addTopping(String
topping) {
14.     toppings.add(topping);
15.   }
16.
17.   public void removeTopping(String
topping) {
18.     toppings.remove(topping|;
19.   }
20. }
```

And:

```
30. class PepperoniPizza extends Pizza {
31.  public void addTopping(String topping) {
32.    System.out.println("Cannot add
Toppings");
33.  }
34.
35.  public void removeTopping(String
topping) {
36.    System.out.println("Cannot remove
Pepperoni");
37.  }
38.}
```

And:

```
50. Pizza pizza = new PepperoniPizza();
51. pizza.addTopping("Mushrooms');
52. pizza.removeTopping('Pepperoni");
```

A. Compilation fails.
B. Cannot add Toppings
C. The code runs with no output.
D. A NullPointerException is thrown in Line 4.

Answer: A

QUESTION: 227

Which three statements are true? (Choose three.)

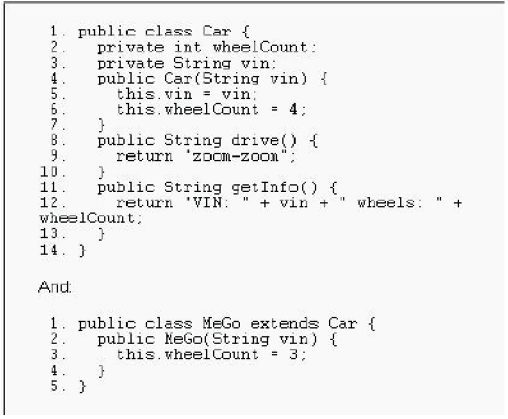
A. A final method in class X can be abstract if and only if X is abstract.
B. A protected method in class X can be overridden by any subclass of X.
C. A private static method can be called only within other static methods in class X.
D. A non-static public final method in class X can be overridden in any subclass of X.
E. A public static method in class X can be called by a subclass of X without explicitly referencing the class X.
F. A method with the same signature as a private final method in class X can be implemented in a subclass of X.
G. A protected method in class X can be overridden by a subclass of X only if the subclass is in the same package as X.

Answer: B,E,F

QUESTION: 228

Click the Exhibit button.

What two must the programmer do to correct the compilation errors? (Choose two.)



```
 1. public class Car {
 2.   private int wheelCount;
 3.   private String vin;
 4.   public Car(String vin) {
 5.     this.vin = vin;
 6.     this.wheelCount = 4;
 7.   }
 8.   public String drive() {
 9.     return "zoom-zoom";
10.   }
11.   public String getInfo() {
12.     return "VIN: " + vin + " wheels: " +
wheelCount;
13.   }
14. }
```

And:

```
 1. public class MeGo extends Car {
 2.   public MeGo(String vin) {
 3.     this.wheelCount = 3;
 4.   }
 5. }
```

A. insert a call to this() in the Car constructor

B. insert a call to this() in the MeGo constructor

C. insert a call to super() in the MeGo constructor

D. insert a call to super(vin) in the MeGo constructor

E. change the wheelCount variable in Car to protected

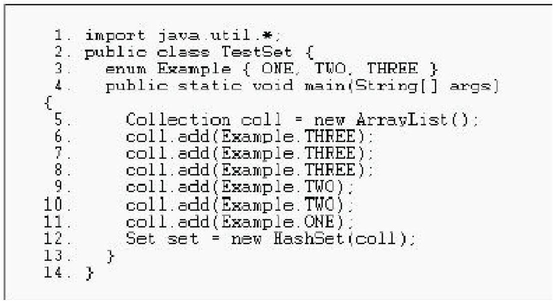
F. change line 3 in the MeGo class to super.wheelCount = 3;

Answer: D,E

QUESTION: 229

Click the Exhibit button.

Which statement is true about the set variable on line 12?



```
 1. import java.util.*;
 2. public class TestSet {
 3.   enum Example { ONE, TWO, THREE }
 4.   public static void main(String[] args)
{
 5.     Collection coll = new ArrayList();
 6.     coll.add(Example.THREE);
 7.     coll.add(Example.THREE);
 8.     coll.add(Example.THREE);
 9.     coll.add(Example.TWO);
10.     coll.add(Example.TWO);
11.     coll.add(Example.ONE);
12.     Set set = new HashSet(coll);
13.   }
14. }
```

A. The set variable contains all six elements from the coll collection, and the order is guaranteed to be preserved.

B. The set variable contains only three elements from the coll collection, and the order is guaranteed to be preserved.

C. The set variable contains all six elements from the coll collection, but the order is NOT guaranteed to be preserved.

D. The set variable contains only three elements from the coll collection, but the order is NOT guaranteed to be preserved.

Answer: D

QUESTION: 230

Given:

```
11. public class Person {
12. private String name, comment;
13. private int age;
14. public Person(String n, int a, String c) {
15. name = n; age = a; comment = c;
16. }
17. public boolean equals(Object o) {
18. if (! (o instanceof Person)) return false;
19, Person p = (Person)o;
20. return age == p.age && name.equals(p.name);
21. }
22. }
```

What is the appropriate definition of the hashCode method in class Person?

A. return super.hashCode();
B. return name.hashCode() + age * 7;
C. return name.hashCode() + comment.hashCode() / 2;
D. return name.hashCode() + comment.hashCode() / 2 - age * 3;

Answer: B

QUESTION: 231

Given:

```
11. public class Key {
12. private long id1;
13. private long id2;
14.
15. // class Key methods
16. }
```

A programmer is developing a class Key, that will be used as a key in a standard java.util.HashMap. Which two methods should be overridden to assure that Key works correctly as a key? (Choose two.)

A. public int hashCode()
B. public boolean equals(Key k)
C. public int compareTo(Object o)
D. public boolean equals(Object o)

E. public boolean compareTo(Key k)

Answer: A,D

QUESTION: 232

Given:

3. import java.util.*;
4. public class Hancock {
5. // insert code here 6. list.add("foo");
7. }
8. }

Which two code fragments, inserted independently at line 5, will compile without warnings? (Choose two.)

A. public void addStrings(List list) {
B. public void addStrings(List<String> list) {
C. public void addStrings(List<? super String> list) {
D. public void addStrings(List<? extends String> list) {

Answer: B,C

QUESTION: 233

A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add(0, object), but does NOT need to support quick random access. What supports these requirements?

A. java.util.Queue
B. java.util.ArrayList
C. java.util.LinearList
D. java.util.LinkedList

Answer: D

QUESTION: 234

Given a class whose instances, when found in a collection of objects, are sorted by using the compareTo() method, which two statements are true? (Choose two.)

A. The class implements java.lang.Comparable.
B. The class implements java.util.Comparator.
C. The interface used to implement sorting allows this class to define only one sort sequence.