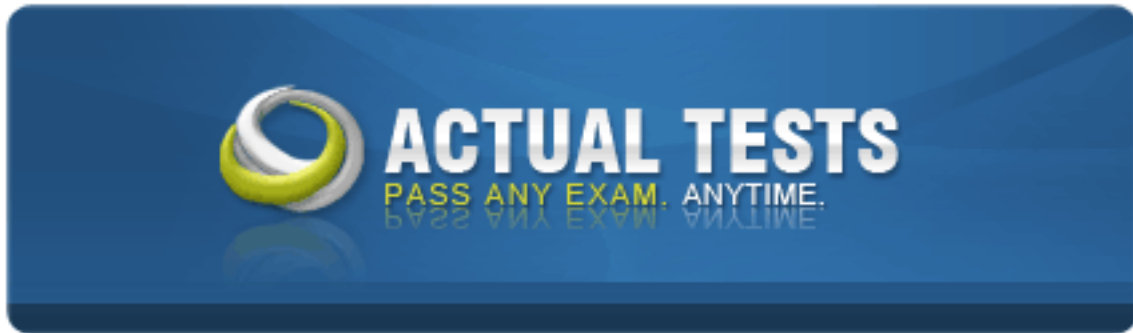


**Oracle 1z0-852**



**Java Standard Edition 6 Programmer Certified  
Professional Upgrade Exam**

**Practice Test**

**Version: 4.0**

**QUESTION NO: 1**

Given:

```
1. public class Base {  
2. public static final String FOO = "foo";  
3. public static void main(String[] args) {  
4. Base b = new Base();  
5. Sub s = new Sub();  
6. System.out.print(Base.FOO);  
7. System.out.print(Sub.FOO);  
8. System.out.print(b.FOO);  
9. System.out.print(s.FOO);  
10. System.out.print(((Base)s).FOO);  
11. } }  
12. class Sub extends Base {public static final String FOO="bar";}
```

What is the result?

- A. foofoofoofoofoo
- B. foobarfoobarbar
- C. foobarfoofoofoo
- D. foobarfoobarfoo
- E. barbarbarbarbar
- F. foofoofoobarbar
- G. foofoofoobarfoo

**Answer: D**

**Explanation:**

**QUESTION NO: 2**

A company has a business application that provides its users with many different reports: receivables reports, payables reports, revenue projects, and so on. The company has just

purchased some new, state-of-the-art, wireless printers, and a programmer has been assigned the task of enhancing all of the reports to use not only the company's old printers, but the new wireless printers as well. When the programmer starts looking into the application, the programmer discovers that because of the design of the application, it is necessary to make changes to each report to support the new printers. Which two design concepts most likely explain this situation? (Choose two.)

- A. Inheritance
- B. Low cohesion
- C. Tight coupling
- D. High cohesion
- E. Loose coupling
- F. Object immutability

**Answer: B,C**

**Explanation:**

### QUESTION NO: 3

Given:

```
5. class Building { }  
6. public class Barn extends Building {  
7. public static void main(String[] args) {  
8. Building build1 = new Building();  
9. Barn barn1 = new Barn();  
10. Barn barn2 = (Barn) build1;  
11. Object obj1 = (Object) build1;  
12. String str1 = (String) build1;  
13. Building build2 = (Building) barn1;  
14. }  
15. }
```

Which is true?

- A. If line 10 is removed, the compilation succeeds.
- B. If line 11 is removed, the compilation succeeds.
- C. If line 12 is removed, the compilation succeeds.
- D. If line 13 is removed, the compilation succeeds.
- E. More than one line must be removed for compilation to succeed.

**Answer: C**

**Explanation:**

#### QUESTION NO: 4

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**

**Explanation:**

**QUESTION NO: 5**

A company that makes Computer Assisted Design (CAD) software has, within its application, some utility classes that are used to perform 3D rendering tasks. The company's chief scientist has just improved the performance of one of the utility classes' key rendering algorithms, and has assigned a programmer to replace the old algorithm with the new algorithm. When the programmer begins researching the utility classes, she is happy to discover that the algorithm to be replaced exists in only one class. The programmer reviews that class's API, and replaces the old algorithm with the new algorithm, being careful that her changes adhere strictly to the class's API. Once testing has begun, the programmer discovers that other classes that use the class she changed are no longer working properly. What design flaw is most likely the cause of these new bugs?

- A. Inheritance
- B. Tight coupling
- C. Low cohesion
- D. High cohesion
- E. Loose coupling
- F. Object immutability

**Answer: B**

**Explanation:**

**QUESTION NO: 6**

Given:

11. class Mammal { }

12.

13. class Raccoon extends Mammal {

14. Mammal m = new Mammal();

15. }

16.

17. class BabyRaccoon extends Mammal { }

Which four statements are true? (Choose four.)

- A. Raccoon is-a Mammal.
- B. Raccoon has-a Mammal.
- C. BabyRaccoon is-a Mammal.
- D. BabyRaccoon is-a Raccoon.
- E. BabyRaccoon has-a Mammal.
- F. BabyRaccoon is-a BabyRaccoon.

**Answer: A,B,C,F**

**Explanation:**

### QUESTION NO: 7

Given:

```
2. public class Hi {  
3. void m1() { }  
4. protected void() m2 { }  
5. } 6. class Lois extends Hi {  
7. // insert code here  
8. }
```

Which four code fragments, inserted independently at line 7, will compile? (Choose four.)

- A. public void m1() { }
- B. protected void m1() { }
- C. private void m1() { }
- D. void m2() { }
- E. public void m2() { }
- F. protected void m2() { }
- G. private void m2() { }

**Answer: A,B,E,F**

**Explanation:**

### QUESTION NO: 8

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**

**Explanation:**

## QUESTION NO: 9

Given the following six method names:

`addListener`

`addMouseListener`

`setMouseListener`

`deleteMouseListener`

`removeMouseListener`

`registerMouseListener`

How many of these method names follow JavaBean Listener naming rules?

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

**Answer: B**

**Explanation:**

#### QUESTION NO: 10

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.
23.     public void testFoo() {
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**

**Explanation:**

#### QUESTION NO: 11

Given:

```
1. class Alligator {  
2. public static void main(String[] args) {  
3. int []x[] = {{1,2}, {3,4,5}, {6,7,8,9}};  
4. int [][]y = x;  
5. System.out.println(y[2][1]);  
6. }  
7. }
```

What is the result?

- A. 2
- B. 3
- C. 4
- D. 6
- E. 7
- F. Compilation fails.

**Answer: E**

**Explanation:**

#### QUESTION NO: 12

Given:

```
11. public static void main(String[] args) {
```

```
12. Object obj = new int[] { 1, 2, 3 };  
13. int[] someArray = (int[])obj;  
14. for (int i : someArray) System.out.print(i + " ");  
15. }
```

What is the result?

- A. 1 2 3
- B. Compilation fails because of an error in line 12.
- C. Compilation fails because of an error in line 13.
- D. Compilation fails because of an error in line 14.
- E. A ClassCastException is thrown at runtime.

**Answer: A**

**Explanation:**

### QUESTION NO: 13

Given:

```
11. public interface A { public void m1(); }  
12.  
13. class B implements A { }  
14. class C implements A { public void m1() { } }  
15. class D implements A { public void m1(int x) { } }  
16. abstract class E implements A { }  
17. abstract class F implements A { public void m1() { } }  
18. abstract class G implements A { public void m1(int x) { } }
```

What is the result?

- A. Compilation succeeds.
- B. Exactly one class does NOT compile.
- C. Exactly two classes do NOT compile.
- D. Exactly four classes do NOT compile.

E. Exactly three classes do NOT compile.

**Answer: C**

**Explanation:**

**QUESTION NO: 14**

Given:

```
21. abstract class C1 {  
22. public C1() { System.out.print(1); }  
23. }  
24. class C2 extends C1 {  
25. public C2() { System.out.print(2); }  
26. }  
27. class C3 extends C2 {  
28. public C3() { System.out.println(3); }  
29. }  
30. public class Ctest {  
31. public static void main(String[] a) { new C3(); }  
32. }
```

What is the result?

- A. 3
- B. 23
- C. 32
- D. 123
- E. 321
- F. Compilation fails.
- G. An exception is thrown at runtime.

**Answer: D**

**Explanation:**

**QUESTION NO: 15**

Given:

```
1. public class A {  
2.     public void doit() {  
3.     }  
4.     public String doit() {  
5.         return "a";  
6.     }  
7.     public double doit(int x) {  
8.         return 1.0;  
9.     }  
10. }
```

What is the result?

- A.** An exception is thrown at runtime.
- B.** Compilation fails because of an error in line 7.
- C.** Compilation fails because of an error in line 4.
- D.** Compilation succeeds and no runtime errors with class A occur.

**Answer: C**

**Explanation:**

**QUESTION NO: 16**

Given:

```
1. public class Plant {  
2.     private String name;  
3.     public Plant(String name) { this.name = name; }  
4.     public String getName() { return name; }
```

5. }

1. public class Tree extends Plant {

2. public void growFruit() { }

3. public void dropLeaves() { }

4. }

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**

**Explanation:**

## QUESTION NO: 17

Given:

11. String test = "Test A. Test B. Test C.";

12. // insert code here

13. String[] result = test.split(regex);

Which regular expression, inserted at line 12, correctly splits test into "Test A", "Test B", and "Test C"?

**A.** String regex = "";

**B.** String regex = " ";

**C.** String regex = ".\*";

**D.** String regex = "\\s";

**E.** String regex = "\\s\\s\*";

**F.** String regex = "\\w[\\.]+";

**Answer: E**

**Explanation:**

**QUESTION NO: 18**

Given that the current directory is empty, and that the user has read and write privileges to the current directory, and the following:

```
1. import java.io.*;
2. public class Maker {
3.     public static void main(String[] args) {
4.         File dir = new File("dir");
5.         File f = new File(dir, "f");
6.     }
7. }
```

Which statement is true?

- A. Compilation fails.
- B. Nothing is added to the file system.
- C. Only a new file is created on the file system.
- D. Only a new directory is created on the file system.
- E. Both a new file and a new directory are created on the file system.

**Answer: B**

**Explanation:**

**QUESTION NO: 19**

Given:

```
22. StringBuilder sb1 = new StringBuilder("123");
23. String s1 = "123";
24. // insert code here
25. System.out.println(sb1 + " " + s1);
```

Which code fragment, inserted at line 24, outputs "123abc 123abc"?

- A. sb1.append("abc"); s1.append("abc");
- B. sb1.append("abc"); s1.concat("abc");
- C. sb1.concat("abc"); s1.append("abc");
- D. sb1.concat("abc"); s1.concat("abc");
- E. sb1.append("abc"); s1 = s1.concat("abc");
- F. sb1.concat("abc"); s1 = s1.concat("abc");
- G. sb1.append("abc"); s1 = s1 + s1.concat("abc");
- H. sb1.concat("abc"); s1 = s1 + s1.concat("abc");

**Answer: E**

**Explanation:**

### QUESTION NO: 20

Given:

- 5. import java.util.Date;
- 6. import java.text.DateFormat;
- 21. DateFormat df
- 22. Date date = new Date();
- 23. // insert code here
- 24. String s = df.format(date);

Which code fragment, inserted at line 23, allows the code to compile?

- A. df = new DateFormat();
- B. df = Date.getFormat();
- C. df = date.getFormat();
- D. df = DateFormat.getFormat();
- E. df = DateFormat.getInstance();

**Answer: E**

**Explanation:**

### QUESTION NO: 21

Given:

```
11. public class Yikes {  
12.  
13. public static void go(Long n) {System.out.print("Long ");}  
14. public static void go(Short n) {System.out.print("Short ");}  
15. public static void go(int n) {System.out.print("int ");}  
16. public static void main(String [] args) {  
17. short y = 6;  
18. long z = 7;  
19. go(y);  
20. go(z); 21. }  
22. }
```

What is the result?

- A. int Long
- B. Short Long
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer: A**

**Explanation:**

## QUESTION NO: 22

Given that c is a reference to a valid java.io.Console object, which two code fragments read a line of text from the console? (Choose two.)

- A. String s = c.readLine();
- B. char[] c = c.readLine();
- C. String s = c.readConsole();
- D. char[] c = c.readConsole();
- E. String s = c.readLine("%s", "name ");
- F. char[] c = c.readLine("%s", "name ");



**Answer: A,E**

**Explanation:**

**QUESTION NO: 23**

A developer is creating a class Book, that needs to access class Paper. The Paper class is deployed in a JAR named myLib.jar. Which three, taken independently, will allow the developer to use the Paper class while compiling the Book class? (Choose three.)

- A.** The JAR file is located at \$JAVA\_HOME/jre/classes/myLib.jar.
- B.** The JAR file is located at \$JAVA\_HOME/jre/lib/ext/myLib.jar..
- C.** The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar/Paper.class.
- D.** The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar.
- E.** The JAR file is located at /foo/myLib.jar and the Book class is compiled using `javac -cp /foo/myLib.jar/Paper Book.java`.
- F.** The JAR file is located at /foo/myLib.jar and the Book class is compiled using `javac -d /foo/myLib.jar Book.java`
- G.** The JAR file is located at /foo/myLib.jar and the Book class is compiled using `javac -classpath /foo/myLib.jar Book.java`

**Answer: B,D,G**

**Explanation:**

**QUESTION NO: 24**

A UNIX user named Bob wants to replace his chess program with a new one, but he is not sure where the old one is installed. Bob is currently able to run a Java chess program starting from his home directory /home/bob using the command:

```
java -classpath /test:/home/bob/downloads/*.jar games.Chess
```

Bob's CLASSPATH is set (at login time) to:

```
/usr/lib:/home/bob/classes:/opt/java/lib:/opt/java/lib/*.jar
```

What is a possible location for the Chess.class file?

- A.** /test/Chess.class
- B.** /home/bob/Chess.class

- C. /test/games/Chess.class
- D. /usr/lib/games/Chess.class
- E. /home/bob/games/Chess.class
- F. inside jarfile /opt/java/lib/Games.jar (with a correct manifest)
- G. inside jarfile /home/bob/downloads/Games.jar (with a correct manifest)

**Answer: C**

**Explanation:**

## QUESTION NO: 25

Given:

```
15. public class Yippee {  
16.     public static void main(String [] args) {  
17.         for(int x = 1; x < args.length; x++) {  
18.             System.out.print(args[x] + " ");  
19.         }  
20.     }  
21. }
```

and two separate command line invocations:

```
java Yippee
```

```
java Yippee 1 2 3 4
```

What is the result?

- A. No output is produced.  
1 2 3
- B. No output is produced.  
2 3 4
- C. No output is produced.  
1 2 3 4
- D. An exception is thrown at runtime.  
1 2 3
- E. An exception is thrown at runtime.

2 3 4

**F.** An exception is thrown at runtime.

1 2 3 4

**Answer: B**

**Explanation:**

### QUESTION NO: 26

Given:

```
11. public class Commander {  
12.     public static void main(String[] args) {  
13.         String myProp = /* insert code here */  
14.         System.out.println(myProp);  
15.     }  
16. }
```

and the command line:

```
java -Dprop.custom=gobstopper Commander
```

Which two, placed on line 13, will produce the output gobstopper? (Choose two.)

- A. System.load("prop.custom");
- B. System.getenv("prop.custom");
- C. System.property("prop.custom");
- D. System.getProperty("prop.custom");
- E. System.getProperties().getProperty("prop.custom");

**Answer: D,E**

**Explanation:**

### QUESTION NO: 27

Given:

```
1. public class Donkey {  
2. public static void main(String[] args) {  
3. boolean assertsOn = false;  
4. assert (assertsOn) : assertsOn = true;  
5. if(assertsOn) {  
6. System.out.println("assert is on");  
7. }  
8. }  
9. }
```

If class Donkey is invoked twice, the first time without assertions enabled, and the second time with assertions enabled, what are the results?

- A. no output
- B. no output  
assert is on
- C. assert is on
- D. no output  
An AssertionError is thrown.
- E. assert is on  
An AssertionError is thrown.

**Answer: D**

**Explanation:**

## QUESTION NO: 28

Given:

```
10. interface Foo {}  
11. class Alpha implements Foo {}  
12. class Beta extends Alpha {}  
13. class Delta extends Beta {  
14. public static void main( String[] args ) {
```

```
15. Beta x = new Beta();  
16. // insert code here  
17. }  
18. }
```

Which code, inserted at line 16, will cause a `java.lang.ClassCastException`?

- A. Alpha a = x;
- B. Foo f = (Delta)x;
- C. Foo f = (Alpha)x;
- D. Beta b = (Beta)(Alpha)x;

**Answer: B**

**Explanation:**

## QUESTION NO: 29

Given:

```
11. public void testIfA() {  
12. if (testIfB("True")) {  
13. System.out.println("True");  
14. } else {  
15. System.out.println("Not true");  
16. }  
17. }  
18. public Boolean testIfB(String str) {  
19. return Boolean.valueOf(str);  
20. }
```

What is the result when method `testIfA` is invoked?

- A. True

- B. Not true
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error at line 12.
- E. Compilation fails because of an error at line 19.

**Answer: A**

**Explanation:**

### QUESTION NO: 30

Which can appropriately be thrown by a programmer using Java SE technology to create a desktop application?

- A. ClassCastException
- B. NullPointerException
- C. NoClassDefFoundError
- D. NumberFormatException
- E. ArrayIndexOutOfBoundsException

**Answer: D**

**Explanation:**

### QUESTION NO: 31

Given:

1. public class Breaker2 {
2. static String o = "";
3. public static void main(String[] args) {
4. z:
5. for(int x = 2; x < 7; x++) {
6. if(x==3) continue;
7. if(x==5) break z;
8. o = o + x;
9. }

```
10. System.out.println(o);  
11. }  
12. }
```

What is the result?

- A. 2
- B. 24
- C. 234
- D. 246
- E. 2346
- F. Compilation fails.

**Answer: B**

**Explanation:**

#### **QUESTION NO: 32**

Given:

```
33. try {  
34. // some code here  
35. } catch (NullPointerException e1) {  
36. System.out.print("a");  
37. } catch (Exception e2) {  
38. System.out.print("b");  
39. } finally {  
40. System.out.print("c");  
41. }
```

If some sort of exception is thrown at line 34, which output is possible?

- A. a
- B. b
- C. c

- D. ac
- E. abc

**Answer: D**

**Explanation:**

**QUESTION NO: 33**

Given:

```
11. public void go(int x) {  
12.     assert (x > 0);  
13.     switch(x) {  
14.         case 2: ;  
15.         default: assert false;  
16.     }  
17. }  
18. private void go2(int x) { assert (x < 0); }
```

Which statement is true?

- A. All of the assert statements are used appropriately.
- B. Only the assert statement on line 12 is used appropriately.
- C. Only the assert statement on line 15 is used appropriately.
- D. Only the assert statement on line 18 is used appropriately.
- E. Only the assert statements on lines 12 and 15 are used appropriately.
- F. Only the assert statements on lines 12 and 18 are used appropriately.
- G. Only the assert statements on lines 15 and 18 are used appropriately.

**Answer: G**

**Explanation:**

**QUESTION NO: 34**

Given:



```
11. public static void main(String[] args) {  
12.     String str = "null";  
13.     if (str == null) {  
14.         System.out.println("null");  
15.     } else (str.length() == 0) {  
16.         System.out.println("zero");  
17.     } else {  
18.         System.out.println("some");  
19.     }  
20. }
```

What is the result?

- A. null
- B. zero
- C. some
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: D**

**Explanation:**

### QUESTION NO: 35

Given:

```
11. class X { public void foo() { System.out.print("X "); } }  
12.  
13. public class SubB extends X {  
14.     public void foo() throws RuntimeException {  
15.         super.foo();  
16.         if (true) throw new RuntimeException();
```

```
17. System.out.print("B ");  
18. }  
19. public static void main(String[] args) {  
20.     new SubB().foo();  
21. }  
22. }
```

What is the result?

- A. X, followed by an Exception.
- B. No output, and an Exception is thrown.
- C. Compilation fails due to an error on line 14.
- D. Compilation fails due to an error on line 16.
- E. Compilation fails due to an error on line 17.
- F. X, followed by an Exception, followed by B.

**Answer: A**

**Explanation:**

### QUESTION NO: 36

Given:

```
3. public class Breaker {  
4.     static String o = "";  
5.     public static void main(String[] args) {  
6.         z:  
7.         o = o + 2;  
8.         for(int x = 3; x < 8; x++) {  
9.             if(x==4) break;  
10.            if(x==6) break z;  
11.            o = o + x;  
12.        }
```

13. `System.out.println(o);`

14. `}`

15. `}`

What is the result?

- A. 23
- B. 234
- C. 235
- D. 2345
- E. 2357
- F. 23457
- G. Compilation fails.

**Answer: G**

**Explanation:**

#### QUESTION NO: 37

Which three will compile and run without exception? (Choose three.)

- A. `private synchronized Object o;`
- B. `void go() {  
synchronized() { /* code here */ }  
}`
- C. `public synchronized void go() { /* code here */ }`
- D. `private synchronized(this) void go() { /* code here */ }`
- E. `void go() {  
synchronized(Object.class) { /* code here */ }  
}`
- F. `void go() {  
Object o = new Object();  
synchronized(o) { /* code here */ }  
}`

**Answer: C,E,F**

**Explanation:**

#### QUESTION NO: 38

Given:

```
11. public class PingPong implements Runnable {  
12.     synchronized void hit(long n) {  
13.         for(int i = 1; i < 3; i++)  
14.             System.out.print(n + "-" + i + " ");  
15.     }  
16.     public static void main(String[] args) {  
17.         new Thread(new PingPong()).start();  
18.         new Thread(new PingPong()).start();  
19.     }  
20.     public void run() {  
21.         hit(Thread.currentThread().getId());  
22.     }  
23. }
```

Which two statements are true? (Choose two.)

- A. The output could be 8-1 7-2 8-2 7-1
- B. The output could be 7-1 7-2 8-1 6-1
- C. The output could be 8-1 7-1 7-2 8-2
- D. The output could be 8-1 8-2 7-1 7-2

**Answer: C,D**

**Explanation:**

#### **QUESTION NO: 39**

Given:

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

3. Thread t = new Thread(new TestOne());

4. t.start();

5. System.out.print("Started");

6. t.join();

7. System.out.print("Complete");

8. }

9. public void run() {

10. for (int i = 0; i < 4; i++) {

11. System.out.print(i);

12. }

13. }

14. }

What can be a result?

**A.** Compilation fails.

**B.** An exception is thrown at runtime.

**C.** The code executes and prints "StartedComplete".

**D.** The code executes and prints "StartedComplete0123".

**E.** The code executes and prints "Started0123Complete".

**Answer: E**

**Explanation:**

### **QUESTION NO: 40 DRAG DROP**

Click the Task button.

Place the code elements into the class so that the code compiles and prints "Run. Run. doIt." in exactly that order. Note that there may be more than one correct solution.

```
public class TestTwo extends Thread {
    public static void main (String[] a) throws Exception {
        TestTwo t = new TestTwo();
        t.start();
        Place here
        Place here
        Place here
    }
    public void run() {
        System.out.print("Run. ");
    }
    public void doIt() {
        System.out.print("doIt. ");
    }
}
```

#### Code Elements

t.start();	t.join();	t.pause(10);	run();	Done
t.run();	t.doIt();	doIt();		

#### Answer:

Place the code elements into the class so that the code compiles and prints "Run. Run. doIt." in exactly that order. Note that there may be more than one correct solution.

```
public class TestTwo extends Thread {
    public static void main (String[] a) throws Exception {
        TestTwo t = new TestTwo();
        t.start();
        t.doIt();
        t.start();
        doIt();
    }
    public void run() {
        System.out.print("Run. ");
    }
    public void doIt() {
        System.out.print("doIt. ");
    }
}
```

#### Code Elements

t.start();	t.join();	t.pause(10);	run();	Done
t.run();	t.doIt();	doIt();		

#### QUESTION NO: 41 DRAG DROP

Click the Task button.

Given:

```

1. import java.util.*;
2. class A { }
3. class B extends A { }
4. public class Test {
5.     public static void main(String[] args) {
6.         List<A> listA = new LinkedList<A>();
7.         List<B> listB = new LinkedList<B>();
8.         List<Object> listO = new LinkedList<Object>();
9.         // insert code here
10.    }
11.    public static void m1(List<? extends A> list) { }
12.    public static void m2(List<A> list) { }
13. }

```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
m1(listA);	m2(listA);	Does not compile.
m1(listB);	m2(listB);	Compiles and runs without error.
m1(listO);	m2(listO);	An exception is thrown at runtime.

Done

**Answer:**

Given:

```

1. import java.util.*;
2. class A { }
3. class B extends A { }
4. public class Test {
5.     public static void main(String[] args) {
6.         List<A> listA = new LinkedList<A>();
7.         List<B> listB = new LinkedList<B>();
8.         List<Object> listO = new LinkedList<Object>();
9.         // insert code here
10.    }
11.    public static void m1(List<? extends A> list) { }
12.    public static void m2(List<A> list) { }
13. }

```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
Compiles and runs without error.	Does not compile.	Does not compile.
Does not compile.	An exception is thrown at runtime.	Compiles and runs without error.
An exception is thrown at runtime.	Compiles and runs without error.	An exception is thrown at runtime.

Done

## QUESTION NO: 42

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**

**Explanation:**

### QUESTION NO: 43

Given a pre-generics implementation of a method:

```
11. public static int sum(List list) {  
12.     int sum = 0;  
13.     for ( Iterator iter = list.iterator(); iter.hasNext(); ) {  
14.         int i = ((Integer)iter.next()).intValue();  
15.         sum += i;  
16.     }  
17.     return sum;  
18. }
```

What three changes allow the class to be used with generics and avoid an unchecked warning? (Choose three.)

- A. Remove line 14.
- B. Replace line 14 with `"int i = iter.next();"`.
- C. Replace line 13 with `"for (int i : intList) {"`.
- D. Replace line 13 with `"for (Iterator iter : intList) {"`.
- E. Replace the method declaration with `"sum(List<int> intList)"`.
- F. Replace the method declaration with `"sum(List<Integer> intList)"`.

**Answer: A,C,F**

**Explanation:**



**QUESTION NO: 44**

Given:

```
12. import java.util.*;

13. public class Explorer2 {

14.     public static void main(String[] args) {

15.         TreeSet<Integer> s = new TreeSet<Integer>();

16.         TreeSet<Integer> subs = new TreeSet<Integer>();

17.         for(int i = 606; i < 613; i++)

18.             if(i%2 == 0) s.add(i);

19.         subs = (TreeSet)s.subSet(608, true, 611, true);

20.         s.add(629);

21.         System.out.println(s + " " + subs);

22.     }

23. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. [608, 610, 612, 629] [608, 610]
- D. [608, 610, 612, 629] [608, 610, 629]
- E. [606, 608, 610, 612, 629] [608, 610]
- F. [606, 608, 610, 612, 629] [608, 610, 629]

**Answer: E**

**Explanation:**

**QUESTION NO: 45**

Given that the elements of a PriorityQueue are ordered according to natural ordering, and:

```
2. import java.util.*;
```

```
3. public class GetInLine {  
4. public static void main(String[] args) {  
5. PriorityQueue<String> pq = new PriorityQueue<String>();  
6. pq.add("banana");  
7. pq.add("pear");  
8. pq.add("apple");  
9. System.out.println(pq.poll() + " " + pq.peek());  
10. }  
11. }
```

What is the result?

- A. apple pear
- B. banana pear
- C. apple apple
- D. apple banana
- E. banana banana

**Answer: D**

**Explanation:**

#### **QUESTION NO: 46**

Given:

```
5. import java.util.*;  
6. public class SortOf {  
7. public static void main(String[] args) {  
8. ArrayList<Integer> a = new ArrayList<Integer>();  
9. a.add(1); a.add(5); a.add(3);  
11. Collections.sort(a);  
12. a.add(2);
```

- 13. Collections.reverse(a);
- 14. System.out.println(a);
- 15. }
- 16. }

What is the result?

- A. [1, 2, 3, 5]
- B. [2, 1, 3, 5]
- C. [2, 5, 3, 1]
- D. [5, 3, 2, 1]
- E. [1, 3, 5, 2]
- F. Compilation fails.
- G. An exception is thrown at runtime.

**Answer: C**

**Explanation:**

#### QUESTION NO: 47

Given:

- 3. import java.util.\*;
- 4. public class G1 {
- 5. public void takeList(List<? extends String> list) {
- 6. // insert code here
- 7. }
- 8. }

Which three code fragments, inserted independently at line 6, will compile? (Choose three.)

- A. list.add("foo");
- B. Object o = list;
- C. String s = list.get(0);
- D. list = new ArrayList<String>();
- E. list = new ArrayList<Object>();

**Answer: B,C,D**

**Explanation:**

**QUESTION NO: 48**

Given:

```
1. import java.util.*;
2.
3. public class LetterASort{
4.     public static void main(String[] args) {
5.         ArrayList<String> strings = new ArrayList<String>();
6.         strings.add("aAaA");
7.         strings.add("AaA");
8.         strings.add("aAa");
9.         strings.add("AAaa");
10.        Collections.sort(strings);
11.        for (String s : strings) { System.out.print(s + " "); }
12.    }
13. }
```

What is the result?

- A. Compilation fails.
- B. aAaA aAa AAaa AaA
- C. AAaa AaA aAa aAaA
- D. AaA AAaa aAaA aAa
- E. aAa AaA aAaA AAaa
- F. An exception is thrown at runtime.

**Answer: C**

**Explanation:**

**QUESTION NO: 49**

Given:

```
1. import java.util.*;
2. public class Example {
3.     public static void main(String[] args) {
4.         // insert code here
5.         set.add(new Integer(2));
6.         set.add(new Integer(1));
7.         System.out.println(set);
8.     }
9. }
```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

- A. Set set = new TreeSet();
- B. Set set = new HashSet();
- C. Set set = new SortedSet();
- D. List set = new SortedList();
- E. Set set = new LinkedHashSet();

**Answer: A**

**Explanation:**

**QUESTION NO: 50 DRAG DROP**

Click the Task button.

Given:

```
1. import java.util.*;
2. public class TestGenericConversion {
3.     public static void main(String[] args) {
4.         List list = new LinkedList();
5.         list.add("one");
6.         list.add("two");
7.         System.out.print(((String)list.get(0)).length());
8.     }
9. }
```

Refactor this class to use generics without changing the code's behavior.

```
1. import java.util.*;
2. public class TestGenericConversion {
3.     public static void main(String[] args) {
4.         Place here
5.         list.add("one");
6.         list.add("two");
7.         Place here
8.     }
9. }
```

Code

List list = new LinkedList();	System.out.print( list.get(0).length() );
List<String> list = new LinkedList<String>();	System.out.print( list.get<String>(0).length() );
List<String> list = new LinkedList();	System.out.print( <String>list.get(0).length() );
List list = new LinkedList<String>();	System.out.print( ((List<String>)list.get(0)).length() );

Done

## Answer:

Given:

```
1. import java.util.*;
2. public class TestGenericConversion {
3.     public static void main(String[] args) {
4.         List list = new LinkedList();
5.         list.add("one");
6.         list.add("two");
7.         System.out.print(((String)list.get(0)).length());
8.     }
9. }
```

Refactor this class to use generics without changing the code's behavior.

```
1. import java.util.*;
2. public class TestGenericConversion {
3.     public static void main(String[] args) {
4.         List<String> list = new LinkedList();
5.         list.add("one");
6.         list.add("two");
7.         System.out.print( list.get<String>(0).length() );
8.     }
9. }
```

Code

List list = new LinkedList();	System.out.print( list.get(0).length() );
List<String> list = new LinkedList<String>();	System.out.print( list.get<String>(0).length() );
List<String> list = new LinkedList();	System.out.print( <String>list.get(0).length() );
List list = new LinkedList<String>();	System.out.print( ((List<String>)list.get(0)).length() );

Done

## QUESTION NO: 51

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.
- D. The interface used to implement sorting allows this class to define many different sort sequences.

**Answer: A,C**

**Explanation:**

## QUESTION NO: 52

Given:

```
12. import java.util.*;

13. public class Explorer3 {

14. public static void main(String[] args) {

15. TreeSet<Integer> s = new TreeSet<Integer>();

16. TreeSet<Integer> subs = new TreeSet<Integer>()

17. for(int i = 606; i < 613; i++)

18. if(i%2 == 0) s.add(i);

19. subs = (TreeSet)s.subSet(608, true, 611, true);

20. subs.add(629);

21. System.out.println(s + " " + subs);

22. }

23. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. [608, 610, 612, 629] [608, 610]
- D. [608, 610, 612, 629] [608, 610, 629]
- E. [606, 608, 610, 612, 629] [608, 610]
- F. [606, 608, 610, 612, 629] [608, 610, 629]

**Answer: B**

**Explanation:**

**QUESTION NO: 53**

Given:

```
23. Object [] myObjects = {  
24. new Integer(12),  
25. new String("foo"),  
26. new Integer(5),  
27. new Boolean(true)  
28. };  
29. Arrays.sort(myObjects);  
30. for(int i=0; i<myObjects.length; i++) {  
31. System.out.print(myObjects[i].toString());  
32. System.out.print(" ");  
33. }
```

What is the result?

- A. Compilation fails due to an error in line 23.
- B. Compilation fails due to an error in line 29.
- C. A ClassCastException occurs in line 29.
- D. A ClassCastException occurs in line 31.
- E. The value of all four objects prints in natural order.

**Answer: C**

**Explanation:**

**QUESTION NO: 54**

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**

**Explanation:**

## QUESTION NO: 55

Given:

- ```
11. // insert code here
12. private N min, max;
13. public N getMin() { return min; }
14. public N getMax() { return max; }
15. public void add(N added) {
16.     if (min == null || added.doubleValue() < min.doubleValue())
17.         min = added;
18.     if (max == null || added.doubleValue() > max.doubleValue())
19.         max = added;
```

20. }

21. }

Which two, inserted at line 11, will allow the code to compile? (Choose two.)

- A. public class MinMax<?> {
- B. public class MinMax<? extends Number> {
- C. public class MinMax<N extends Object> {
- D. public class MinMax<N extends Number> {
- E. public class MinMax<? extends Object> {
- F. public class MinMax<N extends Integer> {

**Answer: D,F**

**Explanation:**

## QUESTION NO: 56

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**

**Explanation:**

**QUESTION NO: 57**

Given:

```
22. public void go() {  
23.     String o = "";  
24.     z:  
25.     for(int x = 0; x < 3; x++) {  
26.         for(int y = 0; y < 2; y++) {  
27.             if(x==1) break;  
28.             if(x==2 && y==1) break z;  
29.             o = o + x + y;  
30.         }  
31.     }  
32.     System.out.println(o);  
33. }
```

What is the result when the go() method is invoked?

- A. 00
- B. 0001
- C. 000120
- D. 00012021
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer: C**

**Explanation:**

**QUESTION NO: 58**

Given a method that must ensure that its parameter is not null:

```
11. public void someMethod(Object value) {
```

12. // check for null value

...

20. System.out.println(value.getClass());

21. }

What, inserted at line 12, is the appropriate way to handle a null value?

**A.** assert value == null;

**B.** assert value != null, "value is null";

**C.** if (value == null) {  
throw new AssertionError("value is null");  
}

**D.** if (value == null) {  
throw new IllegalArgumentException("value is null");  
}

**Answer: D**

**Explanation:**

## QUESTION NO: 59

Given:

11. public class Test {

12. public enum Dogs {collie, harrier, shepherd};

13. public static void main(String [] args) {

14. Dogs myDog = Dogs.shepherd;

15. switch (myDog) {

16. case collie:

17. System.out.print("collie ");

18. case default:

19. System.out.print("retriever ");

20. case harrier:

```
21. System.out.print("harrier ");  
22. }  
23. }  
24. }
```

What is the result?

- A. harrier
- B. shepherd
- C. retriever
- D. Compilation fails.
- E. retriever harrier
- F. An exception is thrown at runtime.

**Answer: D**

**Explanation:**

#### QUESTION NO: 60

Given:

```
5. class A {  
6. void foo() throws Exception { throw new Exception(); }  
7. }  
8. class SubB2 extends A {  
9. void foo() { System.out.println("B "); }  
10. }  
11. class Tester {  
12. public static void main(String[] args) {  
13. A a = new SubB2();  
14. a.foo();  
15. }  
16. }
```

What is the result?

- A. B
- B. B, followed by an Exception.
- C. Compilation fails due to an error on line 9.
- D. Compilation fails due to an error on line 14.
- E. An Exception is thrown with no other output.

**Answer: D**

**Explanation:**

#### QUESTION NO: 61

Given:

```
11. public static void main(String[] args) {  
12. Integer i = new Integer(1) + new Integer(2);  
13. switch(i) {  
14. case 3: System.out.println("three"); break;  
15. default: System.out.println("other"); break;  
16. }  
17. }
```

What is the result?

- A. three
- B. other
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error on line 12.
- E. Compilation fails because of an error on line 13.
- F. Compilation fails because of an error on line 15.

**Answer: A**

**Explanation:**

#### QUESTION NO: 62

Given:

```
1. public class Mule {  
2. public static void main(String[] args) {  
3. boolean assert = true;  
4. if(assert) {  
5. System.out.println("assert is true");  
6. }  
7. }  
8. }
```

Which command-line invocations will compile?

- A. javac Mule.java
- B. javac -source 1.3 Mule.java
- C. javac -source 1.4 Mule.java
- D. javac -source 1.5 Mule.java

**Answer: B**

**Explanation:**

### QUESTION NO: 63

Given:

```
1. public class Donkey2 {  
2. public static void main(String[] args) {  
3. boolean assertsOn = true;  
4. assert (assertsOn) : assertsOn = true;  
5. if(assertsOn) {  
6. System.out.println("assert is on");  
7. }
```

8. }

9. }

If class Donkey is invoked twice, the first time without assertions enabled, and the second time with assertions enabled, what are the results?

**A.** no output

**B.** no output

assert is on

**C.** assert is on

**D.** no output

An AssertionError is thrown.

**E.** assert is on

An AssertionError is thrown.

**Answer: C**

**Explanation:**

#### **QUESTION NO: 64**

Given:

1. public class Venus {

2. public static void main(String[] args) {

3. int [] x = {1,2,3};

4. int y[] = {4,5,6};

5. new Venus().go(x,y);

6. }

7. void go(int[]... z) {

8. for(int[] a : z)

9. System.out.print(a[0]);

10. }

11. }



What is the result?

- A. 1
- B. 12
- C. 14
- D. 123
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer: C**

**Explanation:**

### QUESTION NO: 65

Given:

1. class TestException extends Exception { }
2. class A {
3. public String sayHello(String name) throws TestException {
4. if(name == null) throw new TestException();
5. return "Hello " + name;
6. }
7. }
8. public class TestA {
9. public static void main(String[] args) {
10. new A().sayHello("Aiko");
11. }
12. }

Which statement is true?

- A. Compilation succeeds.
- B. Class A does not compile.
- C. The method declared on line 9 cannot be modified to throw TestException.

D. TestA compiles if line 10 is enclosed in a try/catch block that catches TestException.

**Answer: D**

**Explanation:**

### QUESTION NO: 66

Which two code fragments are most likely to cause a StackOverflowError? (Choose two.)

- A. 

```
int []x = {1,2,3,4,5};  
for(int y = 0; y < 6; y++)  
System.out.println(x[y]);
```
- B. 

```
static int[] x = {7,6,5,4};  
static { x[1] = 8;  
x[4] = 3; }
```
- C. 

```
for(int y = 10; y < 10; y++)  
doStuff(y);
```
- D. 

```
void doOne(int x) { doTwo(x); }  
void doTwo(int y) { doThree(y); }  
void doThree(int z) { doTwo(z); }
```
- E. 

```
for(int x = 0; x < 1000000000; x++)  
doStuff(x);
```
- F. 

```
void counter(int i) { counter(++i); }
```

**Answer: D,F**

**Explanation:**

### QUESTION NO: 67

Given:

1. 

```
class TestA {
```
2. 

```
public void start() { System.out.println("TestA"); }
```
3. 

```
}
```
4. 

```
public class TestB extends TestA {
```
5. 

```
public void start() { System.out.println("TestB"); }
```
6. 

```
public static void main(String[] args) {
```

```
7. ((TestA)new TestB()).start();  
8. }  
9. }
```

What is the result?

- A. TestA
- B. TestB
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer: B**

**Explanation:**

#### QUESTION NO: 68

Given:

```
11. public class Ball{  
12. public enum Color { RED, GREEN, BLUE };  
13. public void foo(){  
14. // insert code here  
15. { System.out.println(c); }  
16. }  
17. }
```

Which code inserted at line 14 causes the foo method to print RED, GREEN, and BLUE?

- A. for( Color c : Color.values() )
- B. for( Color c = RED; c <= BLUE; c++ )
- C. for( Color c ; c.hasNext() ; c.next() )
- D. for( Color c = Color[0]; c <= Color[2]; c++ )
- E. for( Color c = Color.RED; c <= Color.BLUE; c++ )

**Answer: A**

**Explanation:**

**QUESTION NO: 69**

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**

**Explanation:**

**QUESTION NO: 70**

Given:

```
11. public class Rainbow {  
12.     public enum MyColor {  
13.         RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);  
14.         private final int rgb;  
15.         MyColor(int rgb) { this.rgb = rgb; }  
16.         public int getRGB() { return rgb; }  
17.     };  
18.     public static void main(String[] args) {  
19.         // insert code here  
20.     }  
21. }
```

Which code fragment, inserted at line 19, allows the Rainbow class to compile?

- A. MyColor skyColor = BLUE;
- B. MyColor treeColor = MyColor.GREEN;
- C. if(RED.getRGB() < BLUE.getRGB()) { }
- D. Compilation fails due to other error(s) in the code.
- E. MyColor purple = new MyColor(0xff00ff);
- F. MyColor purple = MyColor.BLUE + MyColor.RED;

**Answer: B**

**Explanation:**

## QUESTION NO: 71

Given:

```
10. class Line {  
11.     public class Point { public int x,y;}  
12.     public Point getPoint() { return new Point(); }
```

```
13. }  
14. class Triangle {  
15. public Triangle() {  
16. // insert code here  
17. }  
18. }
```

Which code, inserted at line 16, correctly retrieves a local instance of a Point object?

- A. Point p = Line.getPoint();
- B. Line.Point p = Line.getPoint();
- C. Point p = (new Line()).getPoint();
- D. Line.Point p = (new Line()).getPoint();

**Answer: D**

**Explanation:**

## QUESTION NO: 72

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**

**Explanation:**

### QUESTION NO: 73

Given:

```
11. public class Barn {  
12.     public static void main(String[] args) {  
13.         new Barn().go("hi", 1);  
14.         new Barn().go("hi", "world", 2);  
15.     }  
16.     public void go(String... y, int x) {  
17.         System.out.print(y[y.length - 1] + " ");  
18.     }  
19. }
```

What is the result?

- A. hi hi
- B. hi world
- C. world world
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: D**

**Explanation:**

**QUESTION NO: 74**

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**

**Explanation:**

**QUESTION NO: 75**

Given:



```
15. public class Pass2 {  
16.     public void main(String [] args) {  
17.         int x = 6;  
18.         Pass2 p = new Pass2();  
19.         p.doStuff(x);  
20.         System.out.print(" main x = " + x);  
21.     }  
22.  
23.     void doStuff(int x) {  
24.         System.out.print(" doStuff x = " + x++);  
25.     }  
26. }
```

And the command-line invocations:

```
javac Pass2.java
```

```
java Pass2 5
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. doStuff x = 6 main x = 6
- D. doStuff x = 6 main x = 7
- E. doStuff x = 7 main x = 6
- F. doStuff x = 7 main x = 7

**Answer: B**

**Explanation:**

**QUESTION NO: 76**

Given the following directory structure:

```
bigProject
|--source
|      |--Utils.java
|--classes
|      |--
```

And the following command line invocation:

```
javac -d classes source/Utils.java
```

Assume the current directory is bigProject, what is the result?

- A. If the compile is successful, Utils.class is added to the source directory.
- B. The compiler returns an invalid flag error.
- C. If the compile is successful, Utils.class is added to the classes directory.
- D. If the compile is successful, Utils.class is added to the bigProject directory.

**Answer: C**

**Explanation:**

#### QUESTION NO: 77

A class games.cards.Poker is correctly defined in the jar file Poker.jar. A user wants to execute the main method of Poker on a UNIX system using the command:

```
java games.cards.Poker
```

What allows the user to do this?

- A. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java
- B. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/\*.jar
- C. Put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/Poker.jar
- D. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java
- E. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/\*.jar
- F. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/Poker.jar

**Answer: C**

**Explanation:**

**QUESTION NO: 78**

Click the Exhibit button.

Given the fully-qualified class names:

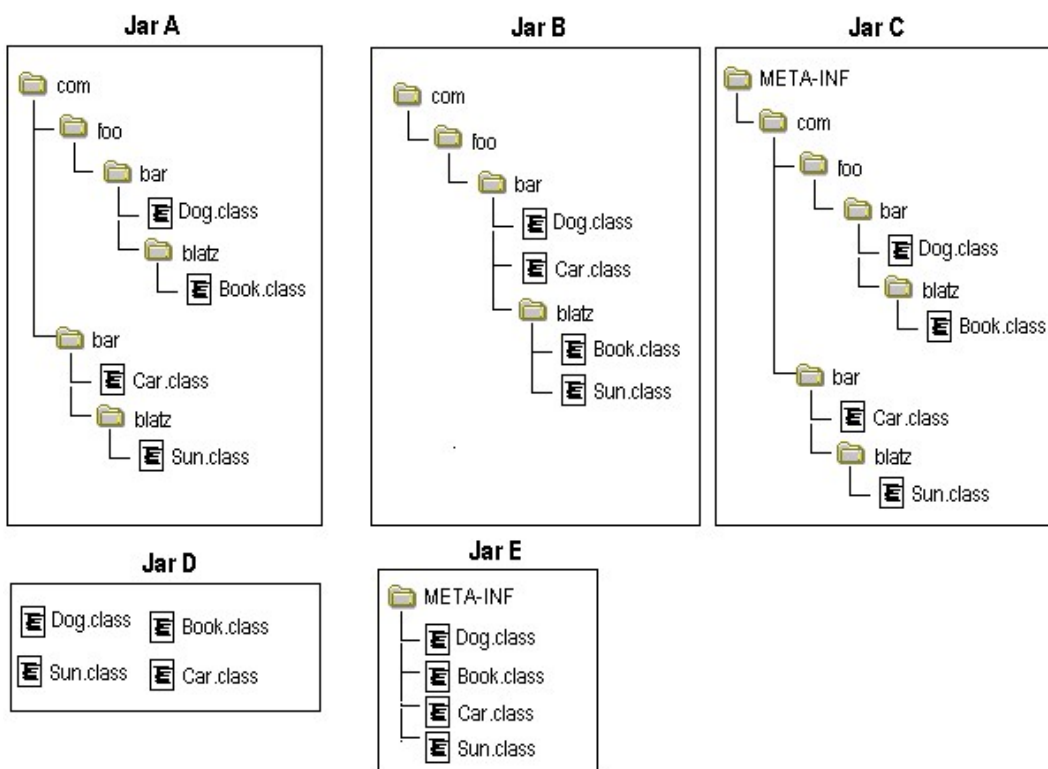
com.foo.bar.Dog

com.foo.bar.blatz.Book

com.bar.Car

com.bar.blatz.Sun

Which graph represents the correct directory structure for a JAR file from which those classes can be used by the compiler and JVM?



- A. Jar A
- B. Jar B
- C. Jar C
- D. Jar D
- E. Jar E

**Answer: A**

**Explanation:**

**QUESTION NO: 79**

Which capability exists only in java.io.FileWriter?

- A. Closing an open stream.
- B. Flushing an open stream.
- C. Writing to an open stream.
- D. Writing a line separator to an open stream.

**Answer: D**

**Explanation:**

**QUESTION NO: 80**

Given a valid DateFormat object named df, and

- 16. Date d = new Date(0L);
- 17. String ds = "December 15, 2004";
- 18. // insert code here

What updates d's value with the date represented by ds?

- A. 18. d = df.parse(ds);
- B. 18. d = df.getDate(ds);
- C. 18. try {  
19. d = df.parse(ds);  
20. } catch(ParseException e) { };
- D. 18. try {  
19. d = df.getDate(ds);  
20. } catch(ParseException e) { };

**Answer: C**

**Explanation:**

**QUESTION NO: 81**

Given:

```
10. import java.io.*;
11. class Animal {
12. Animal() { System.out.print("a"); }
13. }
14. class Dog extends Animal implements Serializable {
15. Dog() { System.out.print("d"); }
16. }
17. 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**

**Explanation:**

**QUESTION NO: 82**

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**

**Explanation:**

**QUESTION NO: 83**

Given:

```
1. public class LineUp {  
2. public static void main(String[] args) {  
3. double d = 12.345;  
4. // insert code here  
5. }  
6. }
```

Which code fragment, inserted at line 4, produces the output | 12.345|?

- A. `System.out.printf("|%7d| \n", d);`
- B. `System.out.printf("|%7f| \n", d);`
- C. `System.out.printf("|%3.7d| \n", d);`
- D. `System.out.printf("|%3.7f| \n", d);`
- E. `System.out.printf("|%7.3d| \n", d);`
- F. `System.out.printf("|%7.3f| \n", d);`

**Answer: F**

**Explanation:**

**QUESTION NO: 84**

Given:

```
1. public class BuildStuff {  
2. public static void main(String[] args) {  
3. Boolean test = new Boolean(true);  
4. Integer x = 343;  
5. Integer y = new BuildStuff().go(test, x);
```

```
6. System.out.println(y);  
7. }  
8. int go(Boolean b, int i) {  
9. if(b) return (i/7);  
10. return (i/49);  
11. }  
12. }
```

What is the result?

- A. 7
- B. 49
- C. 343
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: B**

**Explanation:**

#### **QUESTION NO: 85**

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**

**Explanation:**

#### QUESTION NO: 86

Given that t1 is a reference to a live thread, which is true?

- A. The Thread.sleep() method can take t1 as an argument.
- B. The Object.notify() method can take t1 as an argument.
- C. The Thread.yield() method can take t1 as an argument.
- D. The Thread.setPriority() method can take t1 as an argument.
- E. The Object.notify() method arbitrarily chooses which thread to notify.

**Answer: E**

**Explanation:**

#### QUESTION NO: 87

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**

**Explanation:**

## QUESTION NO: 88

Given:

11. class PingPong2 {
12. synchronized void hit(long n) {
13. for(int i = 1; i < 3; i++)
14. System.out.print(n + "-" + i + " ");
15. }
16. }
17. public class Tester implements Runnable {
18. static PingPong2 pp2 = new PingPong2();

```
19. public static void main(String[] args) {  
20.     new Thread(new Tester()).start();  
21.     new Thread(new Tester()).start();  
22. }  
23. public void run() { pp2.hit(Thread.currentThread().getId()); }  
24. }
```

Which statement is true?

- A. The output could be 5-1 6-1 6-2 5-2
- B. The output could be 6-1 6-2 5-1 5-2
- C. The output could be 6-1 5-2 6-2 5-1
- D. The output could be 6-1 6-2 5-1 7-1

**Answer: B**

**Explanation:**

#### QUESTION NO: 89

Given:

```
1. package test;  
2.  
3. class Target {  
4.     public String name = "hello";  
5. }
```

What can directly access and change the value of the variable name?

- A. any class
- B. only the Target class
- C. any class in the test package
- D. any class that extends Target

**Answer: C**

**Explanation:****QUESTION NO: 90**

Given:

```
11. abstract class Vehicle { public int speed() { return 0; }  
12. class Car extends Vehicle { public int speed() { return 60; }  
13. class RaceCar extends Car { public int speed() { return 150; }  
...  
21. RaceCar racer = new RaceCar();  
22. Car car = new RaceCar();  
23. Vehicle vehicle = new RaceCar();  
24. System.out.println(racer.speed() + ", " + car.speed()  
25. + ", " + vehicle.speed());
```

What is the result?

- A. 0, 0, 0
- B. 150, 60, 0
- C. Compilation fails.
- D. 150, 150, 150
- E. An exception is thrown at runtime.

**Answer: D**

**Explanation:**

**QUESTION NO: 91**

Given:

```
5. class Thingy { Meter m = new Meter(); }  
6. class Component { void go() { System.out.print("c"); } }
```

7. class Meter extends Component { void go() { System.out.print("m"); } }

8.

9. class DeluxeThingy extends Thingy {

10. public static void main(String[] args) {

11. DeluxeThingy dt = new DeluxeThingy();

12. dt.m.go();

13. Thingy t = new DeluxeThingy();

14. t.m.go();

15. }

16. }

Which two are true? (Choose two.)

A. The output is mm.

B. The output is mc.

C. Component is-a Meter.

D. Component has-a Meter.

E. DeluxeThingy is-a Component.

F. DeluxeThingy has-a Component.

**Answer: A,F**

**Explanation:**

## QUESTION NO: 92

Given:

10. interface A { void x(); }

11. class B implements A { public void x() {} public void y() {} }

12. class C extends B { public void x() {} }

And:

20. java.util.List<A> list = new java.util.ArrayList<A>();

```

21. list.add(new B());
22. list.add(new C());
23. for (A a : list) {
24. a.x();
25. a.y();
26. }

```

What is the result?

- A. The code runs with no output.
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 20.
- D. Compilation fails because of an error in line 21.
- E. Compilation fails because of an error in line 23.
- F. Compilation fails because of an error in line 25.

**Answer: F**

**Explanation:**

### QUESTION NO: 93 DRAG DROP

Click the Task button.

Place the Types in one of the Type columns, and the Relationships in the Relationship column, to define appropriate has-a and is-a relationships.

| Type       | Relationship | Type             | Relationships | Types     |
|------------|--------------|------------------|---------------|-----------|
| Place here | Place here   | Animal           | is-a          | Dog       |
| Forest     | Place here   | Place here       | has-a         | Side      |
| Rectangle  | Place here   | Place here       |               | Tail      |
| Place here | Place here   | Programming Book |               | Square    |
|            |              |                  |               | Tree      |
|            |              |                  |               | Book      |
|            |              |                  |               | Java Book |
|            |              |                  |               | Pen       |

Done

**Answer:**

Place the Types in one of the Type columns, and the Relationships in the Relationship column, to define appropriate has-a and is-a relationships.

| Type      | Relationship | Type             | Relationships | Types     |
|-----------|--------------|------------------|---------------|-----------|
| Pen       | Tree         | Animal           | is-a          | Dog       |
| Forest    | Java Book    | Side             | has-a         | Side      |
| Rectangle | is-a         | Dog              |               | Tail      |
| has-a     | Book         | Programming Book |               | Square    |
|           |              |                  |               | Tree      |
|           |              |                  |               | Book      |
|           |              |                  |               | Java Book |
|           |              |                  |               | Pen       |

Done

### QUESTION NO: 94

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**

**Explanation:**

### QUESTION NO: 95

Given:

```
1. class ClassA {  
2. public int numberOfInstances;  
3. protected ClassA(int numberOfInstances) {  
4. this.numberOfInstances = numberOfInstances;  
5. }  
6. }  
7. public class ExtendedA extends ClassA {  
8. private ExtendedA(int numberOfInstances) {  
9. super(numberOfInstances);  
10. }  
11. public static void main(String[] args) {  
12. ExtendedA ext = new ExtendedA(420);  
13. System.out.print(ext.numberOfInstances);  
14. }  
15. }
```

Which statement is true?

- A. 420 is the output.
- B. An exception is thrown at runtime.
- C. All constructors must be declared public.

- D. Constructors CANNOT use the private modifier.
- E. Constructors CANNOT use the protected modifier.

**Answer: A**

**Explanation:**

**QUESTION NO: 96**

Given:

```
21. class Money {  
22.     private String country = "Canada";  
23.     public String getC() { return country; }  
24. }  
25. class Yen extends Money {  
26.     public String getC() { return super.country; }  
27. }  
28. public class Euro extends Money {  
29.     public String getC(int x) { return super.getC(); }  
30.     public static void main(String[] args) {  
31.         System.out.print(new Yen().getC()  
+ " " + new Euro().getC());  
32.     }  
33. }
```

What is the result?

- A. Canada
- B. null Canada
- C. Canada null
- D. Canada Canada
- E. Compilation fails due to an error on line 26.
- F. Compilation fails due to an error on line 29.



**Answer: E**