Given:

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

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
Click the Exhibit button.

```
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. }
```

Given:

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

What is the result?
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

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; }
```

if a variable is static an final the only place where we can initialize is
      a. at the time of declaration
      b. inside static block

```
static final int[] a = { 100,200 };
static{
      a = new int[2];
      a[0] = 100;
      a[1] = 200;
```

```
}
```

Q>

```
1. public class GoTest {
2.    public static void main(String[] args) {
3.          Sente a = new Sente(); a.go();
4.          Goban b = new Goban(); b.go();
5.          Stone c = new Stone(); c.go();
6.    }
7. }
8.
9. class Sente implements Go {
10.   public void go(){
11.         System.out.println("go in Sente");
12. }
13.}
14.
15.class Goban extends Sente {
16.   public void go(){
17.         System.out.println("go in Goban");
18. }
19.
20.}
21.class Stone extends Goban implements Go{
22.}
23.
24.interface Go { public void go(); }
```

What is the result?
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

Q>
What statements are true about the following code? (Choose all that apply.)

```
public class Tail {}
public class Animal {//Animal HAS-A name
       public String name;
}
public class Canine extends Animal {//Canine IS-A Animal,Canine HAS-A Tail,Canine
HAS-A name
       public Tail tail;
}
public class Wolf extends Canine {}//Wolf IS-A Canine,Wolf IS-A Animal,Wolf HAS-A
Tail,Wolf HAS-A name
```

A. Wolf has-a name.
B. Wolf has-a Tail.
C. Wolf is-a Tail.
D. Wolf is-a Animal.
E. Canine is-a Wolf.
F. Animal has-a Tail.

Answer: A,B,D

Q>

Which is a true statement about the following code?

```
public class IsItFurry {
  static interface Mammal { }
  static class Furry implements Mammal { }
  static class Chipmunk extends Furry { }
               public static void main(String[] args) {
                       Chipmunk c = new Chipmunk();
                       Mammal m = c;
                       Furry f = c;
                       int result = 0;
                       if (c instanceof Mammal) result += 1;
                       if (c instanceof Furry) result += 2;
                       if (null instanceof Chipmunk) result += 4;
                       System.out.println(result);
     }
 }
```

A. The output is 0.
B. The output is 3.
C. The output is 7.
D. c instanceof Mammal does not compile.
E. c instanceof Furry does not compile.
F. null instanceof Chipmunk does not compile.

Answer: B

Q>
Which of the following can be inserted in main?

```
public class Outer {
 class Inner { }
       public static void main(String[] args) {
            // INSERT CODE HERE
      }
 }
```

A. Inner in = new Inner();
B. Inner in = Outer.new Inner();
C. Outer.Inner in = new Outer.Inner();
D. Outer.Inner in = new Outer().Inner();
E. Outer.Inner in = new Outer().new Inner();
F. Outer.Inner in = Outer.new Inner();

Answer: E