Q>

```
class Payload {
      private int weight;
      public Payload (int w) { weight = w; }
      public void setWeight(int w) { weight = w; }
      public String toString() { return Integer.toString(weight); }
}
public class TestPayload {
      static void changePayload(Payload p) {
            p.setWeight(420);
            /* insert code */    //Line 12
       }
      public static void main(String[] args) {
                  Payload p = new Payload(200);// weight = 200
                  p.setWeight(1024);//weight = 1024
                  changePayload(p);
                  System.out.println("p is " + p);
      }
}
```

Which code fragment, inserted at the end of line 12, produces the output p is 420?

A. p.setWeight(420);
B. p.changePayload(420);
C. p = new Payload(420);
D. Payload.setWeight(420);
E. p = Payload.setWeight(420);

Answer: A

Q>

```
public static void test(String str) {
      int check = 4;
      if (check = str.length()) {
                  System.out.print(str.charAt(check -= 1) +", ");
      } else {
                  System.out.print(str.charAt(0) + ", ");
      }
}
```

and the invocation:

```
test("four");
test("tee");
test("to");
```

What is the result?

A. r, t, t,
B. r, e, o,
C. Compilation fails.
D. An exception is thrown at runtime.

Answer: C

Question

```
abstract class C1 {
      public C1() { System.out.print(1); }
}
class C2 extends C1 {
      public C2() { System.out.print(2); }
}
class C3 extends C2 {
```

```
      public C3() { System.out.println(3); }
}
public class Ctest {
      public static void main(String[] a) { new C3(); }
}
```

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

Q>
Given:

```
class One {
      public One foo() {
            return this;
      }
}
class Two extends One {
      public One foo() {
            return this;
      }
}
class Three extends Two {
      // insert method here
}
```

Which two methods, inserted individually, correctly complete the Three class?
(Choose two.)

A. public void foo() {}
B. public int foo() { return 3; }
C. public Two foo() { return this; }
D. public One foo() { return this; }
E. public Object foo() { return this; }

Answer : C,D

```
Parent
   | => IS-A relationship
   |
Child
```

Covariant return types
==========

```
Object
   |
String

Number
     |
Integer
```

Q>
Given:

```
public interface A { public void m1(); }

class B implements A { } //CE

class C implements A { public void m1() { } }

class D implements A { public void m1(int x) { } }//CE

abstract class E implements A { }

abstract class F implements A { public void m1() { } }

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

Q>
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"); }//overriden method
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

Q>
Given:

```
class Line {
      public class Point {
             public int x, y;
      }
      public Point getPoint() {
            return new Point();
      }
}
class Triangle {
      public Triangle() {
            // insert code here line 16
      }
}
```

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