Test 2 practice

1. Consider the following class definitions.

```
public class Robot
{
     private int servoCount;
     public int getServoCount()
     {
         return servoCount;
     }
     public void setServoCount(int in)
     {
         servoCount = in;
     }
}
public class Android extends Robot
{
     private int servoCount;
     public Android(int initVal)
     {
         setServoCount(initVal);
     }
     public int getServoCount()
     {
         return super.getServoCount();
     }
     public int getLocal()
     {
         return servoCount;
     }
     public void setServoCount(int in)
     {
         super.setServoCount(in);
     }
     public void setLocal(int in)
     {
         servoCount = in;
     }
}
```

The following code segment appears in a method in another class.

```
int x = 10;
int y = 20;
/* missing code */
```

Which of the following code segments can be used to replace `/* missing code */` so that the value `20` will be printed?

```
    Android a = new Android(x);
(A) a.setServoCount(y);
    System.out.println(a.getServoCount());

    Android a = new Android(x);
(B) a.setServoCount(y);
    System.out.println(a.getLocal());

    Android a = new Android(x);
(C) a.setLocal(y);
    System.out.println(a.getServoCount());

    Android a = new Android(y);
(D) a.setServoCount(x);
    System.out.println(a.getLocal());

    Android a = new Android(y);
(E) a.setLocal(x);
    System.out.println(a.getLocal());
```

2. The `Date` class below will contain three `int` attributes for day, month, and year, a constructor, and a `setDate` method. The `setDate` method is intended to be accessed outside the class.

```
public class Date
{
     /* missing code */
}
```

Which of the following replacements for `/*` *missing code* `*/` is the most appropriate implementation of the class?

Test 2 practice

```
    private int day;
    private int month;
    private int year;
(A) private Date()
    { /* implementation not shown */ }
    private void setDate(int d, int m, int y)
    { /* implementation not shown */ }

    private int day;
    private int month;
    private int year;
(B) public Date()
    { /* implementation not shown */ }
    private void setDate(int d, int m, int y)
    { /* implementation not shown */ }

    private int day;
    private int month;
    private int year;
(C) public Date()
    { /* implementation not shown */ }
    public void setDate(int d, int m, int y)
    { /* implementation not shown */ }

    public int day;
    public int month;
    public int year;
(D) private Date()
    { /* implementation not shown */ }
    private void setDate(int d, int m, int y)
    { /* implementation not shown */ }

    public int day;
    public int month;
    public int year;
(E) public Date()
    { /* implementation not shown */ }
    public void setDate(int d, int m, int y)
    { /* implementation not shown */ }
```

Test 2 practice

3. Consider the following class declarations.

```
public class Base
{
  private int myVal;

  public Base()
  { myVal = 0; }

  public Base(int x)
  { myVal = x; }
}

public class Sub extends Base
{
  public Sub()
  { super(0); }
}
```

Which of the following statements will NOT compile?

(A) `Base b1 = new Base();`
(B) `Base b2 = new Base(5);`
(C) `Base s1 = new Sub();`
(D) `Sub s2 = new Sub();`
(E) `Sub s3 = new Sub(5);`

Test 2 practice

4. Consider the following class definitions.

```
public class Artifact
{
     private String title;
     private int year;
     public Artifact(String t, int y)
     {
         title = t;
         year = y;
     }
     public void printInfo()
     {
         System.out.print(title + " (" + year + ")");
     }
}
public class Artwork extends Artifact
{
     private String artist;
     public Artwork(String t, int y, String a)
     {
         super(t, y);
         artist = a;
     }
     public void printInfo()
     {
         /* missing implementation */
     }
}
```

The following code segment appears in a method in another class.

```
Artwork starry = new Artwork("The Starry Night", 1889, "Van Gogh");
starry.printInfo();
```

The code segment is intended to produce the following output.

```
The Starry Night (1889) by Van Gogh
```

Which of the following can be used to replace `/* missing implementation */` in the `printInfo` method in the `Artwork` class so that the code segment produces the intended output?

(A) `System.out.print(title + " (" + year + ") by " + artist);`

(B) `super.printInfo(artist);`

(C) `System.out.print(super.printInfo() + " by " + artist);`

(D)
```
super();
System.out.print(" by " + artist);
```

(E)
```
super.printInfo();
System.out.print(" by " + artist);
```

Test 2 practice

5. Consider the following three class declarations.

```
public class ClassOne
{
     public void methodA()
     { /* implementation not shown */ }

     public void methodB()
     { /* implementation not shown */ }
}

public class ClassTwo
{
     public void methodA()
     { /* implementation not shown */ }
}

public class ClassThree extends ClassOne
{
     public void methodB()
     { /* implementation not shown */ }
}
```

The following declarations occur in a method in another class.

```
ClassOne one = new ClassOne();
ClassTwo two = new ClassTwo();
ClassThree three = new ClassThree();
/* missing method call */
```

Which of the following replacements for `/* missing method call */` will cause a compile-time error?

(A) `one.methodA();`

(B) `two.methodA();`

(C) `two.methodB();`

(D) `three.methodA();`

(E) `three.methodB();`

Test 2 practice

6. Consider the following class definitions.

```
public class ClassA
{
     public String getValue()
     {
         return "A";
     }
     public void showValue()
     {
         System.out.print(getValue());
     }
}
public class ClassB extends ClassA
{
     public String getValue()
     {
         return "B";
     }
}
```

The following code segment appears in a class other than `ClassA` or `ClassB`.

```
ClassA obj = new ClassB();
obj.showValue();
```

What, if anything, is printed when the code segment is executed?

(A) `A`

(B) `B`

(C) `AB`

(D) `BA`

(E) Nothing is printed because the code does not compile.

7. When designing a class hierarchy, which of the following should be true of a superclass?

(A) A superclass should contain the data and functionality that are common to all subclasses that inherit from the superclass.

(B) A superclass should be the largest, most complex class from which all other subclasses are derived.

(C) A superclass should contain the data and functionality that are only required for the most complex class.

(D) A superclass should have public data in order to provide access for the entire class hierarchy.

(E) A superclass should contain the most specific details of the class hierarchy.

Test 2 practice

8. Consider the following two classes.

```
public class Dog
{
  public void act()
  {
    System.out.print("run ");
    eat();
  }
  public void eat()
  {
    System.out.print("eat ");
  }
}
public class UnderDog extends Dog
{
  public void act()
  {
    super.act();
    System.out.print("sleep ");
  }
  public void eat()
  {
    super.eat();
    System.out.print("bark ");
  }
 }
```

Assume that the following declaration appears in a class other than Dog.

Dog fido = new UnderDog ( ) ;

What is printed as a result of the call fido.act() ?

(A) run eat
(B) run eat sleep
(C) run eat sleep bark
(D) run eat bark sleep
(E) Nothing is printed due to infinite recursion.

9. Consider the following class definition.

```
public class SomeClass
{
     private int x = 0;
     private static int y = 0;
     public SomeClass(int pX)
     {
         x = pX;
         y++;
     }
     public void incrementY()
     { y++; }
     public void incrementY(int inc)
     { y += inc; }
     public int getY()
     { return y; }
}
```

The following code segment appears in a class other than `SomeClass`.

```
SomeClass first = new SomeClass(10);
SomeClass second = new SomeClass(20);
SomeClass third = new SomeClass(30);
first.incrementY();
second.incrementY(10);
System.out.println(third.getY());
```

What is printed as a result of executing the code segment if the code segment is the first use of a `SomeClass` object?

(A) `0`
(B) `1`
(C) `11`
(D) `14`
(E) `30`

Test 2 practice

10. Consider the following class definition.

```
public class Bird
{
     private String species;
     private String color;
     private boolean canFly;
     public Bird(String str, String col, boolean cf)
     {
         species = str;
         color = col;
         canFly = cf;
     }
}
```

Which of the following constructors, if added to the `Bird` class, will cause a compilation error?

(A)
```
public Bird()
{
     species = "unknown";
     color = "unknown";
     canFly = false;
}
```

(B)
```
public Bird(boolean cf)
{
     species = "unknown";
     color = "unknown";
     canFly = cf;
}
```

(C)
```
public Bird(String col, String str)
{
     species = str;
     color = col;
     canFly = false;
}
```

(D)
```
public Bird(boolean cf, String str, String col)
{
     species = str;
     color = col;
     canFly = cf;
}
```

(E)
```
public Bird(String col, String str, boolean cf)
{
     species = str;
     color = col;
     canFly = cf;
}
```