Name(s): _____

CS159 Java Review

Reading Code

1. Which (if any) of the following if-else statements are equivalent?

```
if (a == b)                     if (a == b && b == c)           if (a == b && b == c)
{                                   method1();                      method1();
    if (b == c)                 else                            else if (a == b)
        method1();                  method2();                      method2();
    else
        method2();
}
```

2. What will be printed when the following code executes?

```
for (int i = 0; i < 2; i++)
{
    for(int j = i; j < 3; j++)
    {
        System.out.println(i + " " + j);
    }
}
```

3. Answer the following questions using the class definitions below.

```
public class Sample
{
    public static boolean b1;
    private static boolean b2;
    public boolean b3;
    private  boolean b4;
    // Constructor ommitted.
}
```

```
public class SampleApp
{
    public static void main(String args[])
    {
        Sample s1 = new Sample();
        Sample s2 = new Sample();
    }
}
```

(a) Of the variables b1, b2, b3, b4, which are visible in main?

(b) Of the variables b1, b2, b3, b4, which are associated with <u>objects</u> of the Sample class?

4. What will be printed when the following code executes?

```
public static void main(String[] args)
{
    int num1 = 7;
    int[] numbers = { 1, 2, 3 };

    timesTwo(num1);
    timesTwo(numbers);

    System.out.println("A " + num1);          // Something is printed!
    System.out.println("B " + numbers[0]); // Something is printed!
}

public static void timesTwo(int value)
{
    System.out.println("C " + value);          // Something is printed!
    value = value * 2;
    System.out.println("D " + value);          // Something is printed!
}

public static void timesTwo(int[] values)
{
    for (int i = 0; i < values.length; i++)
    {
        values[i] = values[i] * 2;
    }
}
```

Underline all of the *formal parameters* in the code segment above.

Circle all of the *actual parameters* in the code segment above.

5. What value will be returned if the following method is passed the array {5, 4, 7, 2}?

```
public static int[] r(int[] d)
{// Declare one of the values
int a;
int x = 1;
int    dd = d[0]; // assign d
/* Initialize*/
a = d[0];
while (x   <
  d.length){
 if (d[x]<dd){dd=d[x];}
    if (d[x] > a)
        a = d[x];
        x = 1 + x; // Increment x
        }
int[] c;
c = new int[2];
c[0] = dd; c[1] = a;
return c;}
```

6. What value will be returned if the following method is passed the array {5, 4, 7, 2}?

```
public static int[] minMax(int[] data)
{
    int[] results = new int[2];
    int currentMin = data[0];
    int currentMax = data[0];

    for (int i=1; i < data.length; i++)
    {
        if (data[i] < currentMin)
            currentMin = data[i];

        if (data[i] > currentMax)
            currentMax = data[i];
    }
    results[0] = currentMin;
    results[1] = currentMax;

    return results;
}
```

7. Consider the Point class below:

Point
-xPosition: double
-yPosition: double
+Point(xPosition:double,yPosition:double) +getX(): double +getY(): double +setX(newX:double) +setY(newY:double) +equals(otherPoint:Point): boolean +toString(): String

(a) Is this class mutable or immutable?

(b) Is Point a value or a reference type?

(c) Draw a picture illustrating the contents of memory after the following lines of code are executed.

```
Point a = new Point(2.0, 3.0);
Point b = new Point(4.0, 5.0);
Point c = a;
b = c;
```

Writing Code

1. Write a statement or expression satisfying the requirements below. You may make use of the `Point` class diagrammed above.

   > a) A statement that declares `number` to be a variable of type `float`.
   >
   > b) A statement that instantiates a point at position (0.0, 2.3) and assigns it to the variable `p1`.
   >
   > c) An expression that evaluates to `true` if both `p1` and `p2` contain references to the same point object.
   >
   > d) An expression that evaluates to `true` if `p1` and `p2` are equivalent `Point` objects.
   >
   > e) A statement that declares `points` to be an array of `Points`.
   >
   > f) A statement that instantiates a `Point` array of length 4 , and assigns the result to `points`.
   >
   > g) A for loop that populates the array `points` with four different `Point` objects, each located at position (0,0).

   Draw a picture illustrating the contents of memory after the code in g) is executed.

2. (If time) Provide a Java implementation of the `Point` class diagrammed above.