CS 159 Java Review Exercises   Name(s): _____

The questions below will refer to the `Point` class illustrated in the following UML diagram:

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
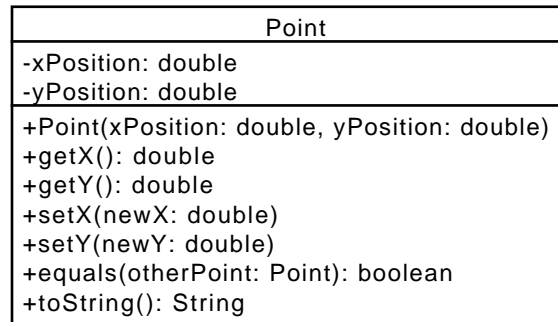
Note that the `toString` method returns the location of the point in the following format: `"(x, y)"`.

1. Is the `Point` class *mutable* or *immutable*?

2. Is `Point` a value or a reference type?

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

   (a) Write a statement that declares `p1` to be a variable of type `Point`.

   (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.

4. 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;
```

5. What will be printed when the following code executes? Draw the contents of all variables as you trace the code.

```
public static void main(String[] args)
{
    Point point1;
    Point point2;

    point1 = new Point(2.0, 3.0);
    point2 = new Point(5.0, 6.0);

    swapPoints(point1, point2);
    System.out.println("A " + point1);
    System.out.println("B " + point2);

    point1 = point2;

    pointTimesTwo(point1);
    System.out.println("C " + point1);
    System.out.println("D " + point2);
}


public static void swapPoints(Point p1, Point p2)
{
    Point tmp;

    tmp = p1;
    p1 = p2;
    p2 = tmp;

    System.out.println("E " + p1);
    System.out.println("F " + p2);
}

public static void pointTimesTwo(Point point)
{
    point.setX(point.getX() * 2.0);
    point.setY(point.getY() * 2.0);
}
```

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

7. 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);
    }
}
```