Practice Exercise #18: Overlapping Rectangles Version 2

http://www.comp.nus.edu.sg/~cs1020/4_misc/practice.html

Objectives:

1. Using **Point** class and **Math** class
2. Creating your own class
3. Problem solving

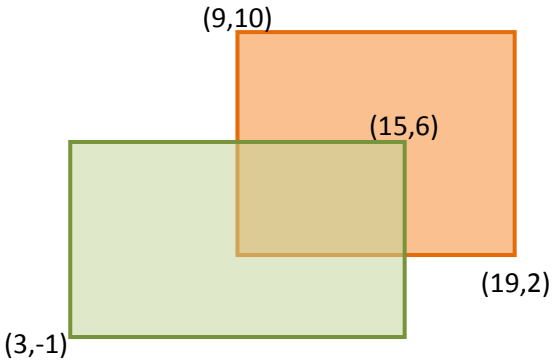
Task statement:

In take-home lab #1, you wrote a program to compute the overlapping area of 2 rectangles. The write-up is given below.

For this exercise, you are to create your own class **MyRect**. The class should contain two attributes **vertex1** and **vertex2** which are the two opposite vertices of a rectangle. You should complete your **MyRect** class with the usual constructors, accessors, mutators, and overriding methods **toString()** and **equals()**.

We can represent a rectangle (whose sides are parallel to the *x*-axis or *y*-axis) with its two opposite vertices, where each vertex is a **Point** object. For example, a rectangle with vertices at (3, -1), (3, 6), (15, 6) and (15, -1) may be represented by any of these 4 pairs of points: (3, -1) and (15, 6); (15, 6) and (3, -1); (3, 6) and (15, -1); or (15, -1) and (3, 6).

You are to write a program **OverlapRectanglesV2.java** to read in integer values for 4 points: the first 2 points are the opposite vertices of one rectangle, and the next 2 points are the opposite vertices of a second rectangle. The figure below shows one rectangle represented by (3, -1) and (15, 6), and another represented by (19, 2) and (9, 10). You should use the **MyRect** class to create the rectangle objects.



Your program should then call the method **overlapArea**(*a*, *b*, *c*, *d*) where *a* and *b* are the opposite vertices of the first rectangle, and *c* and *d* the opposite vertices of the second rectangle. (*a*, *b*, *c*, *d* should be replaced by more descriptive names in your program.) This method computes the overlap area of the two rectangles, which is **24** in our example.

Your program should also print out the rectangles in the format shown below, and check whether the two rectangles are identical. The former makes use of the **toString()** method and the latter the **equals()** method written in **MyRect.java**.

Study the skeleton programs provided. You are to submit both **MyRect.java** and **OverlapRectanglesV2.java**.

Sample run #1:

```
Enter 2 opposite vertices of 1st rectangle: 3 -1 15 6
Enter 2 opposite vertices of 2nd rectangle: 19 2 9 10
1st rectangle: [(3, -1); (15, 6)]
2nd rectangle: [(9, 2); (19, 10)]
They are not identical.
Overlap area = 24
```

Sample run #2:

```
Enter 2 opposite vertices of 1st rectangle: 15 6 3 -1
Enter 2 opposite vertices of 2nd rectangle: 9 2 19 10
1st rectangle: [(3, -1); (15, 6)]
2nd rectangle: [(9, 2); (19, 10)]
They are not identical.
Overlap area = 24
```

Sample run #3:

```
Enter 2 opposite vertices of 1st rectangle: -5 5 -1 1
Enter 2 opposite vertices of 2nd rectangle: 1 2 11 12
1st rectangle: [(-5, 1); (-1, 5)]
2nd rectangle: [(1, 2); (11, 12)]
They are not identical.
Overlap area = 0
```

Sample run #4:

```
Enter 2 opposite vertices of 1st rectangle: -2 -5 8 0
Enter 2 opposite vertices of 2nd rectangle: 8 -5 -2 0
1st rectangle: [(-2, -5); (8, 0)]
2nd rectangle: [(-2, -5); (8, 0)]
They are identical.
Overlap area = 50
```