CS 142 Assignment 5
Colorful Bouncing Circles

Introduction:

In the lecture videos, we wrote the `Circle` class representing circle objects which have a `radius`, `centerX`, and `centerY`. It included `overlaps` and `draw` methods. Then, we wrote a `ColorfulCircle` class which included a `Color` property and overrode the `draw` method of `Circle`.

For this assignment, you are asked to create a `ColorfulBouncingCircle` class.

Unlike the other `Circle` classes, it will include a velocity in the x and y directions. `ColorfulBouncingCircles` will bounce around in a rectangular playing field which the tester class will provide.

Requirements:

Please write a class **`ColorfulBouncingCircle`** as described below. It should **`extend`** `ColorfulCircle`. Begin with the versions of Circle.java, ColorfulCircle.java, and ColorfulBouncingCircleTest.java uploaded to Canvas. DO NOT CHANGE ANY OF THESE CLASSES. These classes, along with the class you create, all need to be in the **same project** in Eclipse.

Your `ColorfulBouncingCircle` class should have private fields for the x and y velocities, and private static fields for the `width` and `height` of the playing field. It should also include these methods:

```
public ColorfulBouncingCircle(double radius, double centerX, double centerY,
Color color, double xVelocity, double yVelocity)
```

This is the constructor. It should call the constructor from the superclass.

```
public static void setPlayingFieldSize(double newWidth, double newHeight)
```

This simple method just sets the size of the playing field that all the circles will move around in. Note that it is static, since there is just one common playing field for all the different circle objects.

```
public void tick()
```

This method will be called by the tester program many times per second, and is how you will implement the animation (think of it like one tick of a clock).

The method will do two things:
If the circle is NOT on the edge of the playing field, then it should alter the circle's center by adding the x and y velocities to their corresponding center coordinates.

If, **after moving**, the new center position would be outside the playing field (either less than zero, or above the width or height), then don't change the center position at all. Instead, alter the velocity to make the circle "bounce" off the wall. If the circle would hit the top or bottom, flip the sign of the y velocity, and if the circle would hit the left or right side, flip the sign of the x velocity. If the circle would hit a corner, flip both velocities.

```
public boolean overlaps(Circle c)
```

This method should override the `overlaps` method from the `Circle` class. It will be used to make the circles bounce off of each other.

First, call the `overlaps` method from the superclass to see if this circle overlaps the other circle.

If they do overlap, then alter this circle's velocity, depending on how it is positioned relative to the other circle:
If the center of this circle is above or below the center of the other circle, flip this circle's y velocity. If this circle's center is to the left or right of the other circle's, then flip this circle's x velocity. As in `tick()`, both may be flipped.

There are several things to note:

- **Please review the velocity changing rules carefully; they are easy to get wrong!!**

- We are treating x as increasing to the right and y as increasing down, as with Java's graphics coordinates system. This was also seen with Circle's draw method.

- The sign flipping will sometimes cause the circles to "vibrate" when caught between each other. This is fine.

- You should not have to write any code which draws; instead, the tester program will animate `ColorfulBouncingCircles` based on your implementation.

- **The test program will ask you to press Enter in the console to launch the automated tests which will assign a tentative score based on your implementation.**

- I am going to run your code using MY `Circle` and `ColorfulCircle` classes, as provided on Canvas. Do not change these! You only need to submit your ColorfulBouncingCircle.java file for grading.