

<b>CS 142 Assignment 5</b> <b>Colorful Bouncing Circle</b> <b>See Canvas for due date!</b>
--

In class, we wrote the Circle class representing circle objects which have a radius, center x, and center y. 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. **Begin with the versions of Circle and ColorfulCircle uploaded to Canvas.**

ColorfulBouncingCircle should **extend** ColorfulCircle. Unlike the other Circle classes, it will include a velocity for the x and y axes. The constructor for ColorfulBouncingCircle includes these parameters:

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

**Note that 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.**

ColorfulBouncingCircles will bounce around in a playing field which the tester will provide. The ColorfulBouncingCircle class should have a public static method to set the playing field width and height:

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

ColorfulBouncingCircle will also include a method which should alter its centerX and centerY with each time tick (similar to a metronome's noise guiding a musician to play a song). The circle's x position should be altered according to x velocity and y position according to y velocity. However, before changing the positions, check if the center position would, after moving, be either less than 0 or greater than the playing field dimensions. If so, instead of changing the center position, alter the velocity. If the ball hits the top or bottom, the sign of its vertical velocity should be flipped; if it hits the left or right, its horizontal velocity sign should be flipped. If it hits a corner, both should be. Notice that velocity may be positive or negative! The tick method which will be called by the tester is defined in this way:

```
public void tick()
```

Finally, we should override the overlaps method to make balls bounce away from each other. You should call the super method to see if your circle overlaps another circle. If so, alter **this** circle's velocity according to its center relative to the other circle. If "this circle" is above or below the "other circle," flip the sign of "this circle's" vertical velocity. If it's to the left or right, flip the horizontal velocity. As before, both may be flipped. **This is NOT the same as a true elastic collision, but it should be simpler for you to implement! The sign flipping will sometimes cause the balls to "vibrate" when caught between each other. Please review the velocity changing rules carefully; they are easy to get wrong!!**

**To review:** Write the ColorfulBouncingCircle class, where you should write the constructor, setPlayingFieldSize, and tick methods, and override the overlaps method, each according to the above descriptions. You should not have to write any code which draws; instead, a test program will be provided which will animate ColorfulBouncingCircles based on your implementation. For every time tick, the test program will compare every circle against every other circle using the overlaps method. **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. DO NOT ALTER the Circle OR ColorfulCircle files!!**