Lab 09 Classes, Turtles and 2D shapes

Shapes - 100 pts

We have been working with a Rectangle class. Build on that to create following functionality.

1. Write a definition for a class named Circle with attributes center and radius, where center is a Point object and radius is a number.
2. Write a function named point\_in\_circle that takes a Circle and a Point and returns True if the Point lies in or on the boundary of the circle.
3. Write a function named rect\_in\_circle that takes a Circle and a Rectangle and returns True if the Rectangle lies entirely in or on the boundary of the circle.
4. Write a function that instantiates 5 Circles objects all with centers at (150, 100) and a radius that is random between 5 and 50. Have your main script call that function, and then print out the center and radius of each as : circle %d : center (%d,%d), radius %d
5. Write a function called draw\_rect that takes a Turtle object and a Rectangle and uses the Turtle to draw the Rectangle. Be sure to call the turtle.mainloop() at the end of your script to keep the drawing visible.
6. Write a function called draw\_circle that takes a Turtle and a Circle and draws the Circle
7. Use your circle class and turtle to draw the olympics rings with their proper colors and a penwidth of 10. Look at the following python docs page for guidance on the turtle functions and setting attributes like pencolor and pensize.

[https://docs.python.org/3.2/library/turtle.html#overview-of-available-turtle-and-screen-methods](https://docs.python.org/3.2/library/turtle.html" \l "overview-of-available-turtle-and-screen-methods)

10 bonus pts if you make the olympic rings plot a function, that takes an origin point, a circle radius and a turtle as arguments. (It can just use home as the origin). The test code for this will look like the following:

P1 = Point(-50,0)

olympics(P1,25,bob\_the\_turtle)

P2 = Point(-250,0)  
olympics(P2,100,bob\_the\_turtle)