Project-Chapter 10 (100 points)

Student Name: Jared Robbins NETID: jlrob

**Project: The Circle2D Class**

Problem Description:

Define the Circle2D class that contains:

* Two double data fields named x and y that specify the center of the circle with get methods.
* A data field radius with a get method.
* A no-arg constructor that creates a default circle with (0, 0) for (x, y) and 1 for radius.
* A constructor that creates a circle with the specified x, y, and radius.
* A method getArea() that returns the area of the circle.
* A method getPerimeter() that returns the perimeter of the circle.
* A method contains(double x, double y) that returns true if the specified point (x, y) is inside this circle. See Figure 10.14(a).
* A method contains(Circle2D circle) that returns true if the specified circle is inside this circle. See Figure 10.14(b).
* A method overlaps(Circle2D circle) that returns true if the specified circle overlaps with this circle. See the figure below.

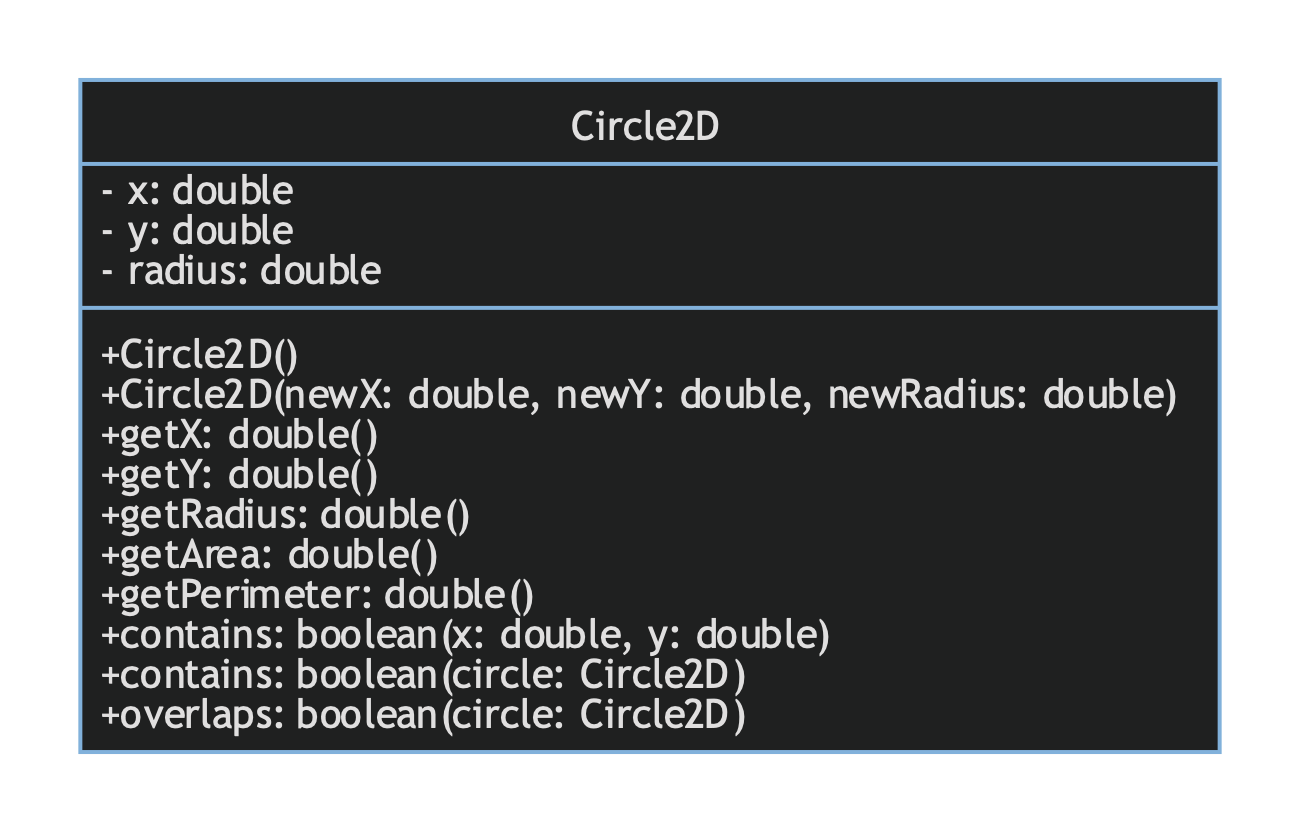
(a) (b) (c)

Figure

(a) A point is inside the circle. (b) A circle is inside another circle. (c) A circle overlaps another circle.

Draw the UML diagram for the class. Implement the class. Write a test program that creates a Circle2D object c1 (new Circle2D(2, 2, 5.5)), displays its area and perimeter, and displays the result of c1.contains(3, 3), c1.contains(new Circle2D(4, 5, 10.5)), and c1.overlaps(new Circle2D(3, 5, 2.3)).

Design:



Coding: (Copy and Paste Source Code here. Format your code using Courier 10pts)

*import* org.jetbrains.annotations.NotNull;  
  
*/\*\* \_\_Adapted From\_\_: `Project-Chapter10.doc`, provided as part of coursework; 2022-11-12  
 \* \_Project-Chapter10\_: The 'Circle2D' Class; 22FA-CSC501B-OL-14927 (Graduate Programming Practicum)  
 \* \_Date\_: 2022-11-12  
 \* \_Author\_: Jared Robbins  
 \* \_NETID\_: jlrob  
 \* \_Description\_: A program that tests the implementation of a 'Circle2D' class, with no user input.  
 \* Program compares relative positioning of specified instances of the 'Circle2D' class to determine if  
 \* an instance contains the origin (midpoint) of another, if an instance is contained within another instance  
 \* or if two instances overlap each other.  
 \* The program determines this with methods built in to Circle2D and then outputs the results. \*/*

*public class* Exercise10\_11 {  
 *public static void* main(String[] args) {  
 Circle2D c1 = *new* Circle2D(2, 2, 5.5);  
 System.out.println("Area is " + c1.getArea());  
 System.out.println("Perimeter is " + c1.getPerimeter());  
 System.out.println(c1.contains(3, 3));  
 System.out.println(c1.contains(*new* Circle2D(4, 5, 10.5)));  
 System.out.println(c1.overlaps(*new* Circle2D(3, 5, 2.3)));  
 }  
}  
*class* Circle2D {  
 *private double* x = 0;  
 *private double* y = 0;  
 *private double* radius = 1;  
 *public* Circle2D() {}  
 *public* Circle2D(*double* newX, *double* newY, *double* newRadius) {  
 *this*.x = newX;  
 *this*.y = newY;  
 *this*.radius = newRadius;  
 }  
 *public double* getX() {*return this*.x;}  
 *public double* getY() {*return this*.y;}  
 *public double* getRadius() {*return this*.radius;}  
 *public double* getArea() {*return* (radius \* radius) \* Math.PI;}  
 *public double* getPerimeter() {*return* 2 \* Math.PI \* radius;}

*public boolean* contains(*double* x, *double* y) {  
 *double* distance = Math.*pow*((x - *this*.x), 2) + Math.*pow*((y - *this*.y), 2);  
 *return* Math.*sqrt*(distance) <= radius;  
 }  
 *public boolean* contains(@NotNull Circle2D circle) {  
 *double* distance = Math.*pow*((circle.x - *this*.x), 2) + Math.*pow*((circle.y - *this*.y), 2);  
 *if* (Math.*sqrt*(distance) <= *this*.radius) {  
 *return* circle.radius < *this*.radius - Math.*sqrt*(distance);  
 } *else return false*;  
 }  
 *public boolean* overlaps(@NotNull Circle2D circle) {  
 *double* distance = Math.*pow*((circle.x - *this*.x), 2) + Math.*pow*((circle.y - *this*.y), 2);  
 *return* distance > circle.radius + *this*.radius;  
 }  
}

Submit the following items:

1. Submit this word file to Blackboard before the due day

2. Submit the source code files to Blackboard before the due day