1. (The ***MyPoint*** class from Exercise 10.4) Design a class named **MyPoint** to represent a point with **x**- and **y**-coordinates. the class contains:

* The data fields x and y that represent the coordinates with getter methods.
* A no-arg constructor that creates a point (**0**, **0**).
* A constructor that constructs a point with specified coordinates.
* A method named **distance** that returns the distance from this point to a specified point of the **MyPoint** type.
* A method named **distance** that returns the distance from this point to another point with specified **x**- and **y**-coordinates.

1. (the Cricle2D class ***modified*** from Exercise 10.11) Define the **Circle2D** class that contains:

* A field named **center** of the type **MyPoint** that represents the center of the circle, with a getter method.
* A field **radius** with a getter method.
* A no-arg constructor that creates a default circle with center (**0**, **0**) and **1** for **radius**.
* A constructor that creates a circle with the specified **center** 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(MyPoint p)** that returns **true** if the specified point **p** is inside this circle (see Figure 10.21a on Page 403).
* A method **contains (Circle2D circle)** that returns **true** if the specified circle is inside this circle (see Figure 10.21b on Page 403).
* A method **overlaps(Circle2D circle)** that returns **true** if the specified circle overlaps with this circle (see Figure 10.21c on Page 403).

1. Write a test program that:

* Creates 4 points: **p1(2, 2)**, **p2(3, 3)**, **p3(4, 5)**,and **p4(3, 5)**.
* Creates a Circle2D object **c1 (new Circle2D(p1, 5.5))**, displays its area and perimeter.
* Displays the result of **c1.contains(p2)**.
* Displays the result of **c1.contains(new Circle2D(p3, 10.5))**.
* Displays the result of **c1.overlaps(new Circle2D(p4, 2.3))**.