LAB 6

Exercise 1: Create a Class & Methods

- Create a circle class that
  - contains attributes radius (double), xcoor (double), ycoor (double), and a constant pi (double) from Math class (Math.PI)
  - has the following method
    - setRadius(double r)
    - double getRadius()
    - double getArea()
    - double getPerimeter()
    - double getDiameter()
    - setCenter(double x, double y)
    - double getAreaOfCircles(int n) //return the area of n circles
    - printInfo() // print the information of a circle (radius, center coordinates, area, perimeter, and diameter)

Exercise 1: Create a Class & Methods

- Restrict the access of class attributes by setting them private

- Allow all methods to be public

- The Circle class (completed)

```
public class Circle{
            private double radius;
            private final double PI = Math.PI;
            private double xcoor;
            private double ycoor;
            public void setRadius(double r){
                        radius = r;
            }
            public double getRadius(){
                        return radius;
            }
            public double getArea(){
                        double area = PI*radius*radius;
                        return area;
            }
            public double getPerimeter(){
                        return 2*PI*radius;
            }

            public double getAreaOfCircles(int n){
                        return n*getArea();
            }
            public double getDiameter(){
                        return 2*radius;
            }
            public void setCenter(int x, int y){
                        xcoor = x;
                        ycoor = y;
            }
            public void printInfo(){
                        System.out.println("The circle have:");
                        System.out.println("radius = "+radius);
                        System.out.println("area = "+getArea());
                        System.out.println("area = "+getPerimeter());
                        System.out.println("diameter = "+getDiameter());
                        System.out.println("coordinate  = ("+xcoor+","+ycoor+")");
            }
}
```

Exercise 2: Create a Driver Class

- Create a driver class named “CircleCaller”

```
public class CircleCaller{

        public static void main(String args){

        }

}
```

- Inside the main method:
  - Create a Circle object called c1
  - Set radius = 2.5
  - Set xcoor = 0 and ycoor = 8
  - Then print info of c1

Exercise 2: Create a Driver Class

```
public class CircleCaller{
	public static void main(String[] args){
		Circle c1 = new Circle();
		c1.setRadius(2.5);
		c1.setCenter(0, 8);
		c1.printInfo();
	}

}
```

- Run and observe the result

Exercise 3: Create a Constructor

- Modify your Circle class by adding a static attribute call circleCounter (int)
- Set the attribute for public access
- Add (override) a default constructor to allow the circleCounter increase every time an object of a Circle is created

- The Circle class

```
public class Circle{
        private double radius;
        private final double PI = Math.PI;
        private double xcoor;
        private double ycoor;
        public staticint circleCounter;

        public Circle(){
                circleCounter++;
        }

        public void setRadius(double r){
                radius = r;
        }
        public double getRadius(){
                return radius;
        }
        public double getArea(){
                double area = PI*radius*radius;
                return area;
        }
        ...
        ...
        ...


}
```

Exercise 3: Create a Constructor

- Modify your CircleCaller class by
  - printing the static attribute circleCounter
  - creating a new Circle object called c2 and then printing the static attribute circleCounter
  - creating another new Circle object called c3 and then printing the static attribute circleCounter again

Exercise 3: Create a Constructor

```
public class CircleCaller{
        public static void main(String[] args){
                Circle c1 = new Circle();
                c1.setRadius(2.5);
                c1.setCenter(0, 8);
                c1.printInfo();
                System.out.println(Circle.circleCounter);
                Circle c2 = new Circle();
                System.out.println(Circle.circleCounter);
                Circle c3 = new Circle();
                System.out.println(Circle.circleCounter);
        }

}
```

- Run and observe the result

Exercise 4: Create another Constructor

- Modify your Circle class so that it allows three ways for you to create a Circle object as follows:
  - Creating an object without setting anything at the beginning
    - We've already implement this with the default constructor
  - Creating an object and set the radius at the beginning of object creation
  - Creating an object and set both radius and center at the beginning
  - Adding a method to get center coordinator as well

- The Circle class

```
public class Circle{
          ...
          ...
          private double ycoor;
          public static int circleCounter;

          public Circle(){
                    circleCounter++;
          }
          public Circle(double r){
                    radius = r;
                    circleCounter++;
          }
          public Circle(double r, double x, double y){
                    radius = r;
                    xcoor = x;
                    ycoor = y;
                    circleCounter++;
          }

          public void setRadius(double r){
                    radius = r;
          }
          ...
          ...
          ...
          public String getCenter(){
                    return xcoor + "," + ycoor;
          }

}
```

Exercise 4: Create another Constructor

- Modify your CircleCaller class by:
  - Creating c1 without setting any parameters
  - Creating c2 with the radius of 5.3 by using the second constructor
  - Creating c3 with the radius of 1.2, xcoor of 4, ycoor of 10

Exercise 4: Create another Constructor

```
public class CircleCaller{
          public static void main(String[] args){
                    Circle c1 = new Circle();
                    System.out.println("Number of Circle created so far: " + Circle.circleCounter);
                    System.out.println("---c1---");
                    System.out.println("Radius: "+c1.getRadius()+"\t Center: "+c1.getCenter());

                    System.out.println("---c2---");
                    Circle c2 = new Circle(5.3);
                    System.out.println("Number of Circle created so far: " + Circle.circleCounter);
                    System.out.println("Radius: "+c2.getRadius()+"\t Center: "+c2.getCenter());

                    System.out.println("---c3---");
                    Circle c3 = new Circle(1.2, 4, 10);
                    System.out.println("Number of Circle created so far: " + Circle.circleCounter);
                    System.out.println("Radius: "+c3.getRadius()+"\t Center: "+c3.getCenter());
          }

}
```

- Run and observe the result

Exercise 4: this Reference

- Modify your Circle class by:
  - Changing the parameter name of the constructor Circle (double r) from r to radius
  - Run the driver class and observe result

- The Circle class

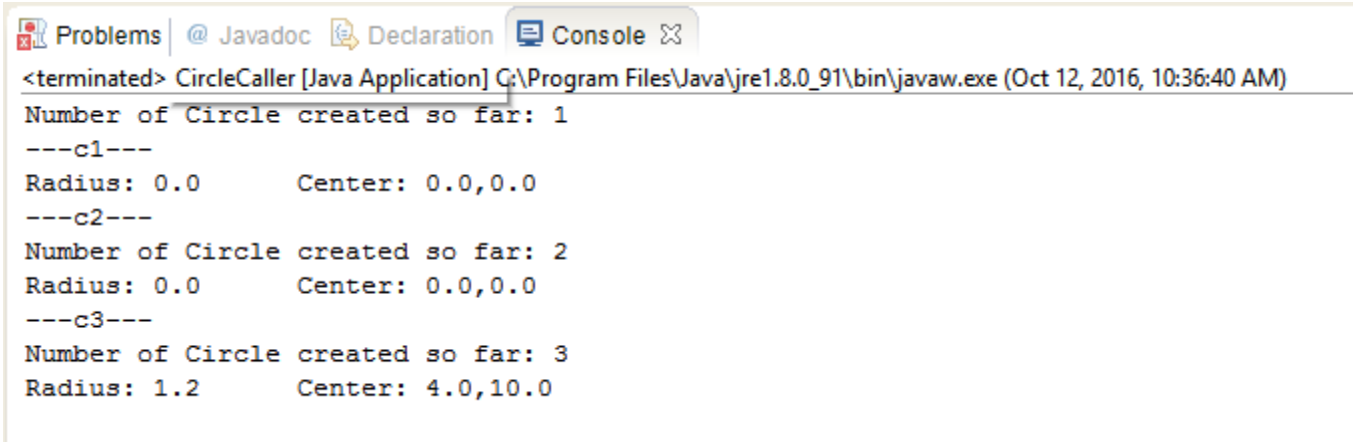
```
public class Circle{
          ...
          ...
          private double ycoor;
          public static int circleCounter;

          public Circle(){
                    circleCounter++;
          }
          public Circle(double radius){
                    radius = radius;
                    circleCounter++;
          }
          public Circle(double r, double x, double y){
                    radius = r;
                    xcoor = x;
                    ycoor = y;
                    circleCounter++;
          }

          public void setRadius(double r){
                    radius = r;
          }
          ...
          ...
          ...
          public String getCenter(){
                    return xcoor + "," + ycoor;
          }

}
```

Exercise 4: this Reference



- The radius of c2 is 0.0 why?

Exercise 4: this Reference

- Modify your Circle class by:
  - Using this reference in the constructor

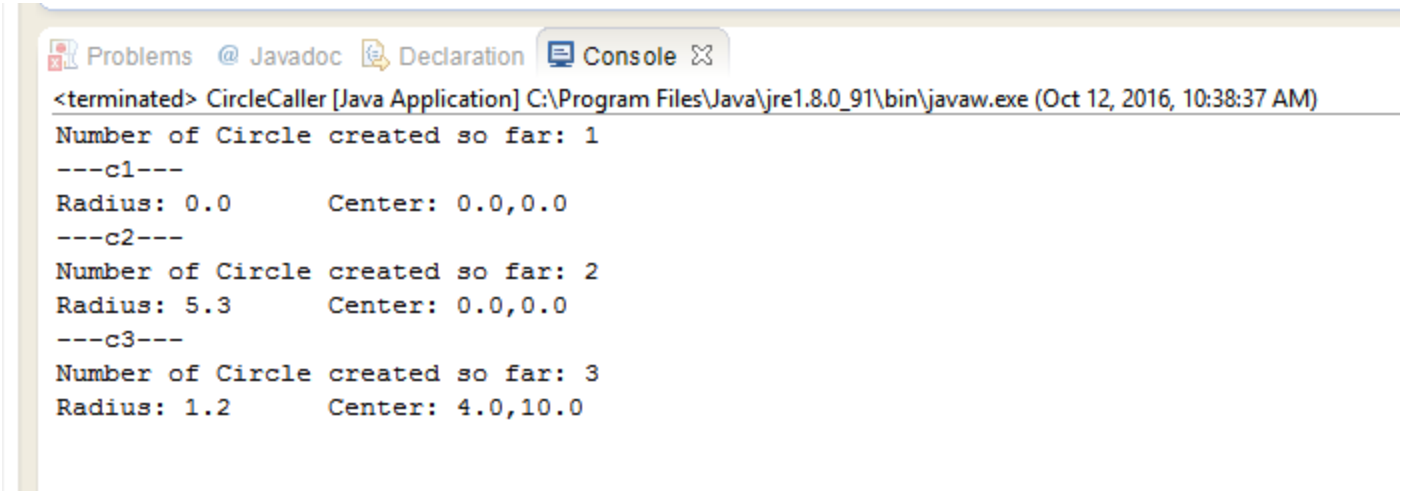
```
public class Circle{
          ...
          ...
          private double ycoor;
          public static int circleCounter;

          public Circle(){
                    circleCounter++;
          }
          public Circle(double radius){
                    this.radius = radius;
                    circleCounter++;
          }
          ...
          ...
          ...
          public String getCenter(){
                    return xcoor + "," + ycoor;
          }

}
```

Exercise 4: this Reference

- Run the driver class and observe result



- The radius of c2 is now set to 5.3 properly, why?