```java
//Brandon London 03/20/2019//
import java.util.Scanner;

public class Exercise_10 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("Enter a, b, c: ");
        double a = input.nextDouble();
        double b = input.nextDouble();
        double c = input.nextDouble();

        QuadraticEquation equation = new QuadraticEquation(a, b, c);
        double discriminant = equation.getDiscriminant();

        if (discriminant > 0) {
            System.out.println("The roots are " + equation.getRoot1()
                    + " and " + equation.getRoot2());
        } else if (discriminant == 0) {
            System.out.println("The root is " + equation.getRoot1());
        } else {
            System.out.println("The equation has no roots");

        }
    }

```

Problems @ Javadoc Declaration Console

```
<terminated> Exercise_10 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Mar 20,
Enter a, b, c: 1 2 3
The equation has no roots
```

```
<terminated> Exercise_10 [Java Application] C:\Program Files\Java\jre1.8.0_18
Enter a, b, c: 1.0 3 1
The roots are -0.3819660112501051 and -2.618033988749895
```

```
Enter a, b, c: 1 2.0 1
The root is -1.0
```

```java
//Brandon London 03/20/2019//
// UML diagram
/*****************************************************
*                 QuadraticEquation                  *
------------------------------------------------------
* -a: double                                         *
* -b: double                                         *
* -c: double                                         *
* +QuadradticEquation(a: double, b: double, c: double) *
* +getA(): double                                    *
* +getB(): double                                    *
* +getC(): double                                    *
* +getDiscriminant(): double                         *
* +getRoot1(): double                                *
* +getRoot2(): double                                *
*****************************************************/
public class QuadraticEquation {

    private double a;
    private double b;
    private double c;

    public QuadraticEquation(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public double getA() {
        return a;
    }

    public void setA(double a) {
        this.a = a;
    }

    public double getB() {
```

```java
    public double getA() {
        return a;
    }

    public void setA(double a) {
        this.a = a;
    }

    public double getB() {
        return b;
    }

    public void setB(double b) {
        this.b = b;
    }

    public double getC() {
        return c;
    }

    public void setC(double c) {
        this.c = c;
    }

    public double getDiscriminant() {
         return b * b - 4.0 * a * c;
    }

    public double getRoot1() {
            return  (-b + Math.pow(getDiscriminant(), 0.5)) / (2.0 * a);
    }

    public double getRoot2() {
            return  (-b - Math.pow(getDiscriminant(), 0.5)) / (2.0 * a);
    }

}
```

```
//Brandon London 03/20/2019//
/**************************************************************************
 * (Geometry: the Circle2D class) Define the Circle2D class that contains:*
 *                                                                        *
 * ■ Two double data fields named x and y that specify the center of the circle *
 *   with getter methods.                                                 *
 * ■ A data field radius with a getter 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 10.21a).    *
 * ■ A method contains(Circle2D circle) that returns true if the specified circle *
 *   is inside this circle (see Figure 10.21b).                           *
 * ■ A method overlaps(Circle2D circle) that returns true if the specified circle *
 *   overlaps with this circle (see Figure 10.21c).                       *
 *                                                                        *
 * Draw the UML diagram for the class and then 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)).                                                  *
 **************************************************************************/
```

Problems | Javadoc | Declaration | Console

```
<terminated> Exercise_11 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Mar 20, 2019, 11:28:01 AM)
Circle1 area: 95.03317777109125
Circle1 perimeter: 34.55751918948772
Does circle1 contain the point (3, 3)? true
Does circle1 contain the circle centered at (4, 5) and radius 10.5? true
Does circle1 overlap the circle centered at (3, 5) and radius 2.3? true
```

```
public class Exercise_11 {
    /** Main method */
    public static void main(String[] args) {
        // Create a Circle2D object
        Circle2D c1 = new Circle2D(2, 2, 5.5);

        // Display results
        System.out.println("Circle1 area: " + c1.getArea());
        System.out.println("Circle1 perimeter: " + c1.getPerimeter());
        System.out.println(
            "Does circle1 contain the point (3, 3)? " + c1.contains(3, 3));
        System.out.println(
            "Does circle1 contain the circle centered at (4, 5) and radius 10.5? "
            + c1.contains(new Circle2D(4, 5, 10.5)));
        System.out.println(
            "Does circle1 overlap the circle centered at (3, 5) and radius 2.3? "
            + c1.overlaps(new Circle2D(3, 5, 2.3)));
    }
}
```

```java
//Brandon London 03/20/2019//
/*************************************************
*                    Circle2D                    *
*------------------------------------------------*
* -x: double                                     *
* -y: double                                     *
* -radius: double                                *
* +Circle2D()                                    *
* +Circle2D(x: double, y: double, radius: double) *
* +getX(): double                                *
* +getY(): double                                *
* +getRadius(): double                           *
* +getArea(): double                             *
* +getPerimeter(): double                        *
* +contains(x: double, y: double): boolean       *
* +contains(Circle2D: circle): boolean           *
* +overlaps(Circle2D: circle): boolean           *
*************************************************/

// Implement Circle2D class
public class Circle2D {
    /**  Data fields */
    private double x;
    private double y;
    private double radius;
```

Problems @ Javadoc Declaration Console

<terminated> Exercise_11 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Mar

```
Circle1 area: 95.03317777109125
Circle1 perimeter: 34.55751918948772
Does circle1 contain the point (3, 3)? true
Does circle1 contain the circle centered at (4, 5) and radius 10.5? true
Does circle1 overlap the circle centered at (3, 5) and radius 2.3? true
```

```java
    private double radius;

    /** Create a default Circle2D with
    *  (0,0) for (x,y) and 1 for radius */
    Circle2D() {
        this(0, 0, 1);
    }

    /** Create a Circle2D with specified x,y, and radius */
    Circle2D(double x, double y, double radius) {
        this.x = x;
        this.y = y;
        this.radius = radius;
    }

    /** Return x */
    public double getX() {
        return x;
    }

    /** Return y */
    public double getY() {
        return y;
    }

    /** Return radius */
    public double getRadius() {
        return radius;
    }

    /** Return the area of the circle */
    public double getArea() {
        return Math.PI * Math.pow(radius, 2);
    }

    /** Return the perimeter of the circle */
    public double getPerimeter() {
```

```
52          return radius;
53      }
54
55      /** Return the area of the circle */
56      public double getArea() {
57          return Math.PI * Math.pow(radius, 2);
58      }
59
60      /** Return the perimeter of the circle */
61      public double getPerimeter() {
62          return 2 * Math.PI * radius;
63      }
64
65      /** Return true if the specified point
66      *   (x, y) is inside this circle     */
67      public boolean contains(double x, double y) {
68          return Math.sqrt(Math.pow(x - this.x, 2) +
69                  Math.pow(y - this.y, 2))
70                  < radius;
71      }
72
73      /** Return true if the specified
74      *   circle is inside this circle */
75      public boolean contains(Circle2D circle) {
76          return Math.sqrt(Math.pow(circle.getX() - x, 2) +
77                  Math.pow(circle.getY() - y, 2))
78                  <= Math.abs(radius - circle.getRadius());
79      }
80
81      /** Return true if the specified
82      *   circle overlaps with this circle */
83      public boolean overlaps(Circle2D circle) {
84          return Math.sqrt(Math.pow(circle.getX() - x, 2) +
85                  Math.pow(circle.getY() - y, 2))
86                  <= radius + circle.getRadius();
87      }
88  }
```