```
/*A class called MyPoints, which models a 2D point with x and y coordinates,
is designed as follows:
• Two instance variables x (int) and y (int).
• A default (or "no-arg") constructor that construct a point at the default
location of (0, 0).
• A overloaded constructor that constructs a point with the given x and y
coordinates.
• A method setXY() to set both x and y.
• A method getXY() which returns the x and y in a 2-element int array.
• A toString() method that returns a string description of the instance in
the format "(x, y)".
• A method called distance(int x, int y) that returns the distance from this
point to another point at the
given (x, y) coordinates
• An overloaded distance(MyPoint another) that returns the distance from this
point to the given
MyPoint instance (called another)
• Another overloaded distance() method that returns the distance from this
point to the origin (0,0)
Develop the code for the class MyPoint. Also develop a JAVA program (called
TestMyPoint) to test all the methods defined in the class.
4th Program from the List*/
public class MyPoints {
    int x;
    int y;

    // Default constructor with no arguments
    public MyPoints() {
       this.x = 0;
       this.y = 0;
    }
    // Overloaded constructor with two arguments
    public MyPoints(int m, int n) {
        this.x = m;
        this.y = n;
    }
    // Method to set both x and y
    public void setXY(int m, int n) {
        this.x = m;
        this.y = n;
    }
    // Method to get x and y in a 2-element int array
    public int[] getXY() {
        int[] coordinates = {this.x, this.y};
        return coordinates;
    }

 // Method to return a string description of the instance
    public String toString() {
```

```
        return "(" + this.x + ", " + this.y + ")";
    }

// Method to calculate distance to another point with
//given coordinates
    public double distance(int m, int n) {
       int Xdiff = this.x - m;
       int Ydiff = this.y - n;
       return Math.sqrt(Xdiff * Xdiff + Ydiff * Ydiff);
    }

// Overloaded method to calculate distance to another
//MyPoint instance
    public double distance(MyPoints another) {
        return distance(another.x, another.y);
    }

// Overloaded method to calculate distance to the
//origin (0,0)
    public double distance() {
        return distance(0, 0);
    }
// MyPoints class is tested in the main() method

public static void main(String[] args) {
// Test the MyPoints class no-arg & args constructors
        MyPoints point1 = new MyPoints();
        MyPoints point2 = new MyPoints(3, 4);

// Initial points are displayed
//compiler uses point1.toString(),point2.toString()
//toString()returns string description of the instance

        System.out.println("Point 1: " + point1);
        System.out.println("Point 2: " + point2);

// setXY method tested
  point1.setXY(5, 6);
  System.out.println("Point 1 after setXY: " + point1);

// getXY method tested
int[] coordinates = point2.getXY();
System.out.println("Coordinates of Point 2: (" +
coordinates[0] + ", " + coordinates[1] + ")");
```

```
// distance methods tested
System.out.println("Distance between Point 1 and (0, 0): "
+ point1.distance());
System.out.println("Distance between Point 1 and Point 2: "
+ point1.distance(point2));
System.out.println("Distance between Point 1 and (3, 4): "
+ point1.distance(3, 4));
    }
}
```