

CS 1073

FR03A

Assignment #3

Ethan A. McCarthy

3573807

Section 1

```
import java.util.Scanner;
/**
 * @author Ethan McCarthy 3573807
 */

public class TabletPicker{
    public static void main(String[] args){
        Scanner read = new Scanner(System.in);
        boolean answer;
        //initialize the scanner and the answer variable

        //asks the user to input yes or no and start asking the questions
        System.out.println("(Answer \"yes\" or \"no\" to the following
questions.)");
        System.out.println("Is the larger screen necessary for your job?");
        String input = read.nextLine();

        //answer variable is true if the input is yes
        answer = input.equals("yes");

        //if answer is true tell the user to buy the tablet if not keep asking
questions
        if(answer == true){
            System.out.println("Buy the 13 inch tablet.");
        }
        else{
            //keep asking questions
            System.out.println("Will you watch a lot of content on your
tablet?");
            input = read.nextLine();
            answer = input.equals("yes");

            if (answer == true){
                System.out.println("Buy the 13 inch tablet.");
            }
            else{
                System.out.println("Will you draw on your tablet frequently?");
                input = read.nextLine();
                answer = input.equals("yes");


                if(answer == false){
                    System.out.println("You should buy the 11 inch tablet.");
                }
            }
        }
    }
}
```

```
        else{
            //last question
            System.out.println("Does the extra screen space matter to you
when drawing?");

            input = read.nextLine();
            answer = input.equals("yes");

            if (answer == true){
                System.out.println("You should buy the 13 inch tablet.");
            }
            else{
                System.out.println("You should buy the 11 inch tablet.");
            }
        }
    }
}
}
```

Section 2

 Command Prompt

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\barym>cd C:\Users\barym\Desktop\CS 1073\asgn 3

C:\Users\barym\Desktop\CS 1073\asgn 3>javac TabletPicker.java

C:\Users\barym\Desktop\CS 1073\asgn 3>java TabletPicker
(Answer "yes" or "no" to the following questions.)
Is the larger screen necessary for your job?
no
Will you watch a lot of content on your tablet?
no
Will you draw on your tablet frequently?
yes
Does the extra screen space matter to you when drawing?
yes
You should buy the 13 inch tablet.

C:\Users\barym\Desktop\CS 1073\asgn 3>
```

```
C:\Users\barym\Desktop\CS 1073\asgn 3>java TabletPicker
(Answer "yes" or "no" to the following questions.)
Is the larger screen necessary for your job?
yes
Buy the 13 inch tablet.

C:\Users\barym\Desktop\CS 1073\asgn 3>_
```

```
C:\Users\barym\Desktop\CS 1073\asgn 3>java TabletPicker
(Answer "yes" or "no" to the following questions.)
Is the larger screen necessary for your job?
no
Will you watch a lot of content on your tablet?
yes
Buy the 13 inch tablet.

C:\Users\barym\Desktop\CS 1073\asgn 3>_
```

Section 3

```
/**
 * @author Ethan McCarthy 3573807
 */

public class LineSegment{

    //endpoint 1
    private CartesianPoint endpoint1;

    //endpoint 2
    private CartesianPoint endpoint2;

    /**
     * constructor for the line segment
     * @param point1 first point
     * @param point2 second point
     */
    public LineSegment(CartesianPoint point1, CartesianPoint point2){
        endpoint1 = point1;
        endpoint2 = point2;
    }

    /**
     * another line constructor using x and y values instead of a cartesian point
     * @param xVal1
     * @param yVal1
     * @param xVal2
     * @param yVal2
     */
    public LineSegment(double xVal1, double yVal1, double xVal2, double yVal2){
        endpoint1 = new CartesianPoint(xVal1, yVal1);
        endpoint2 = new CartesianPoint(xVal2, yVal2);
    }

    /**
     * method to get endpoint 1
     * @return first endpoint
     */
    public CartesianPoint getEndpoint1() {
        return endpoint1;
    }

    /**
```

```

    * method to get endpoint 2
    * @return second endpoint
    */
    public CartesianPoint getEndpoint2() {
        return endpoint2;
    }

    /**
     * retrieve info about the line segment
     * @return information in a string
     */
    public String toString(){
        return "Line Segment:\n\tStarting Point: " + endpoint1 + "\n\tEndpoint: "
+ endpoint2;
    }

    /**
     * method to calculate the length
     * @return the length
     */
    public double length(){
        return endpoint1.distance(endpoint2);
    }

    /**
     * method to check if the line crosses an axis
     * @return true or false
     */
    public boolean crossesAxis(){
        if ((endpoint1.getX() < 0 && endpoint2.getX() > 0) || (endpoint1.getX() >
0 && endpoint2.getX() < 0)){
            return true;
        }
        else if((endpoint1.getY() < 0 && endpoint2.getY() > 0) ||
(endpoint1.getY() > 0 && endpoint2.getY() < 0)){
            return true;
        }
        else{
            return false;
        }
    }

    /**
     * method to calculate the slope of the line

```

```

    * @return the slope
    */
    public double slope(){
        if((endpoint2.getX() - endpoint1.getX()) == 0){
            return Double.POSITIVE_INFINITY;
        }
        else{
            double slope = (endpoint2.getY() - endpoint1.getY()) / (endpoint2.getX()
- endpoint1.getX());
            return slope;
        }
    }

    /**
     * method to see if the given point is on the line or not
     * @param pointIn given point
     * @return true or false
     */
    public boolean containsPoint(CartesianPoint pointIn){
        //y=mx+b
        double x = pointIn.getX();
        double y = pointIn.getY();
        double TOLERANCE = 1e-14;

        //getting b
        double yInt = y - (slope() * x);

        //checks to see if its on the line segment
        if (Math.abs((slope()*x + yInt) - y) < TOLERANCE){
            return true;
        }
        else{
            return false;
        }
    }
}

```

Section 4

```
/**
 * @author Ethan McCarthy 3573807
 */

public class LineSegmentTest{

    public static void main(String[] args){

        CartesianPoint point1 = new CartesianPoint(1, 4);
        CartesianPoint point2 = new CartesianPoint(-5, -10);

        LineSegment line1 = new LineSegment(point1, point2);
        LineSegment line2 = new LineSegment(1, 30, 2, -37);

        CartesianPoint p1 = new CartesianPoint(1, 4);
        CartesianPoint p2 = new CartesianPoint(-100, 100);

        System.out.println("The slope of line segment 1 is: " + line1.slope());
        System.out.println("The length of line segment 1 is: " + line1.length() +
"\n");

        System.out.println("The slope of line segment 2 is: " + line2.slope());
        System.out.println("The length of line segment 2 is: " + line2.slope() +
"\n");

        if(line1.crossesAxis() == true){
            System.out.println("The first line segment crosses at least one
axis.");
        }
        else{
            System.out.println("The first line does not cross an axis");
        }

        if(line2.crossesAxis() == true){
            System.out.println("The second line crosses at least one axis.");
        }
        else{
            System.out.println("The second line does not cross an axis.");
        }

        if(line1.containsPoint(p1) == true){
            System.out.println("The point p1 is on the first line.");
        }
    }
}
```



```
    else{
        System.out.println("The point p1 is not on the line.");
    }

    if(line1.containsPoint(p2) == true){
        System.out.println("The point p2 is on the line.");
    }
    else{
        System.out.println("The point p2 is not on the line.");
    }
}
}
```

Section 5

The slope of line segment 1 is: 2.333333333333335

The length of line segment 1 is: 15.231546211727817

The slope of line segment 2 is: -67.0

The length of line segment 2 is: -67.0

The first line segment crosses at least one axis.

The second line crosses at least one axis.

The point p1 is on the first line.

The point p2 is not on the line.