**CS1073  
FR03B**

**Assignment #3**

**Daniyal Khan  
3765942**

**Question I:**

1. **Source code:**

/\*\*

@author Daniyal Khan 3765942

\*/

import java.util.Scanner;

public class Luna {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

String userInput;

// Initial Question 1

System.out.println("Is there an animal in the yard?");

userInput = scan.nextLine();

if(userInput.equalsIgnoreCase("Yes")) {

System.out.println("Is it moving?"); // Question 2

userInput = scan.nextLine();

if(userInput.equalsIgnoreCase("Yes")) {

System.out.println("Is it smaller than me?"); // Question 3

userInput = scan.nextLine();

if(userInput.equalsIgnoreCase("Yes")) {

System.out.println("Is she Prof.Bidlake in an online meeting?"); // Question 4

userInput = scan.nextLine();

if(userInput.equalsIgnoreCase("Yes")) {

System.out.println("Bark");

}

else if (userInput.equalsIgnoreCase("No")) {

System.out.println("Don't bark");

}

}

else if (userInput.equalsIgnoreCase("No")) {

System.out.println("Bark");

}

}

else if (userInput.equalsIgnoreCase("No")) {

System.out.println("Don't bark");

}

}

else if (userInput.equalsIgnoreCase("No")) {

System.out.println("Don't bark");

}

else {

System.out.println("Invalid input. Please enter 'Yes' or 'No'.");

}

}

}

1. **Output:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Question II:**

1. **Source code:**

/\*\*

This class represents a triangle shape using 3 points.

@author Natalie Webber

@author Daniyal Khan 3765942

\*/

public class Triangle {

private CartesianPoint pointA;

private CartesianPoint pointB;

private CartesianPoint pointC;

public Triangle (double x1, double y1,

double x2, double y2,

double x3, double y3) {

pointA = new CartesianPoint (x1, y1);

pointB = new CartesianPoint (x2, y2);

pointC = new CartesianPoint (x3, y3);

}

public Triangle (CartesianPoint p1,

CartesianPoint p2,

CartesianPoint p3) {

pointA = p1;

pointB = p2;

pointC = p3;

}

/\*\*

This method returns the perimeter of a triangle

@return The perimeter of the triangle

\*/

public double getPerimeter() {

return pointA.distance(pointB) + pointB.distance(pointC) + pointC.distance(pointA);

}

/\*\*

This method tells if a triangle is equilateral or not

@return True if the triangle is an equilateral and false if not

\*/

public boolean isEquilateral() {

double distanceAb = pointA.distance(pointB);

double distanceBc = pointB.distance(pointC);

double distanceCa = pointC.distance(pointA);

double tolerance = 1e-14;

if ((Math.abs(distanceAb-distanceBc) < tolerance)

&& Math.abs(distanceBc-distanceCa) < tolerance) {

return true;

}

return false;

}

/\*\*

This method tells if a triangle is right-angled or not

@return True if the triangle is right-angled and false if not

\*/

public boolean isRight() {

double distanceAb = pointA.distance(pointB);

double distanceBc = pointB.distance(pointC);

double distanceCa = pointC.distance(pointA);

double tolerance = 1e-14;

double sqDistanceAb = distanceAb \* distanceAb;

double sqDistanceBc = distanceBc \* distanceBc;

double sqDistanceCa = distanceCa \* distanceCa;

if (Math.abs(sqDistanceAb + sqDistanceBc - sqDistanceCa) < tolerance ||

Math.abs(sqDistanceBc + sqDistanceCa - sqDistanceAb) < tolerance ||

Math.abs(sqDistanceCa + sqDistanceAb - sqDistanceBc) < tolerance) {

return true;

}

return false;

}

}

1. **Source code for the driver:**

/\*\*

@author Daniyal Khan 3765942

\*/

public class TestTriangle {

public static void main(String[] args) {

Triangle t1 = new Triangle(-0.5, 0.0, 0.5, 0.0, 0.0, Math.sqrt(3)/2);

Triangle t2 = new Triangle(0.0, 0.0, 1.0, 0.0, 0.0, 1.0);

if(t1.isEquilateral()) {

System.out.println("The triangle t1 is a equilateral triangle");

}

else {

System.out.println("The triangle t1 is not an equilateral triangle");

}

if(t2.isRight()) {

System.out.println("The triangle t2 is a right angle triangle");

}

else {

System.out.println("The triangle t2 is not a right angle triangle");

}

}

}

**Output:**

The triangle t1 is a equilateral triangle

The triangle t2 is a right angle triangle

A screenshot of a computer

Description automatically generated