

# Lab 1: Triangles

Professor: Ronaldo Menezes

TA: Ivan Bogun

Department of Computer Science  
Florida Institute of Technology

August 17, 2014

## 1 Problem statement

Given a set of  $N$  points on a plane find distinct points which make up triangles of the smallest and the largest areas.

## 2 Implementation

Implement the class *TriangleProblem.java*.

---

```
// TriangleProblem.java
public class TriangleProblem {

    double[][] points;

    // constructor. Nothing to implement here
    public TriangleProblem(double[][] points_) {
        this.points=points_;
    }

    public Triangle calculateLowestAreaTriangle() {
        // implement the function which would find triangle of the smallest
        // area. Triangle is made from 3 distinct points.
    }

    public Triangle calculateHighestAreaTriangle() {
        // implement the function which would find triangle of the largest area.
    }
}
```

---

Implement the class *Triangle.java* which would serve as a data structure for triangles.

---

```
// Triangle.java
public class Triangle {

    double[] a;                // three points which make up a
    // triangle. Each points is an array of size 2
    double[] b;
    double[] c;

    public String toString(){
        // return string-like representation of the triangle. Example will be
        // given in the input-output section
    }

    public double calculateArea(){
```

```
// calculate the area of the triangle and return it
}
}
```

---

### 3 Sample input-output

Create the file *Driver.java* whose modified version will be used for testing.

#### 3.1 Input

---

```
import java.util.Random;

public class Driver {
    public static void main(String[] args) {
        // In order to generate SAME numbers each time use fixed seed
        // value
        int seed=1;
        Random random = new Random(seed);

        // number of points to generate
        int n=200;

        double[][] points=new double[n][2];

        // generate input
        for (int i = 0; i < n; i++) {
            points[i][0]=random.nextDouble()*100;
            points[i][1]=random.nextDouble()*100;
        }

        TriangleProblem triangleProblem = new TriangleProblem(points);

        double start=0,end=0;
        Triangle solution;

        System.out.println("Smallest area triangle:");

        // time the function
        start=System.currentTimeMillis();
        solution=triangleProblem.calculateLowestAreaTriangle();
        end=System.currentTimeMillis();

        System.out.println(solution.toString());
        System.out.println((end-start)/1000+" seconds");
    }
}
```

```
        System.out.println("Largest area triangle:");
        start=System.currentTimeMillis();
        solution=triangleProblem.calculateHighestAreaTriangle();
        end=System.currentTimeMillis();

        System.out.println(solution.toString());
        System.out.println((end-start)/1000+" seconds");

    }

}
```

---

### 3.2 Output

```
Smallest area triangle:
Point A: 3.237285340092644 66.97852634820785
Point B: 5.687004880600954 55.984109541990776
Point C: 6.5946891002912515 51.911086899415125
Area: 8.477417007925291E-4
0.871 seconds
Largest area triangle:
Point A: 96.77559094241207 0.6117182265761301
Point B: 0.5025175992452557 52.313515578833304
Point C: 96.62539506822094 99.65234515119386
Area: 4763.590071608187
0.93 seconds
```

## 4 Grade breakdown

basis	grade
Implementation	(60)
Triangle class	20
Lowest area	20
Highest area	20
Comments	(10)
General	10
Overall	(30)
Compiled	15
Style	5
Runtime	10
Total	100