SCSJ1023, Programming Technique II
Semester 2, 2017/2108

Lab Exercise 2
Class and Object Manipulations

A point can be described by its coordinates *x* and *y*. The distance between two points, $p1 = (x_1, y_1)$ and $p2 = (x_2, y_2)$, is calculated by the phytogras theorem as given below:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

If the vertices (or points) of a polygon are $p1, p2, p3 \ldots, pn$ (as shown in Figure1), the center of the polygon, *c*, can be determined by the average of all the points of the polygon as given below:

$$c = \frac{p1 + p2 + p3 + \cdots + pn}{n}$$

where *n* is the number of points.

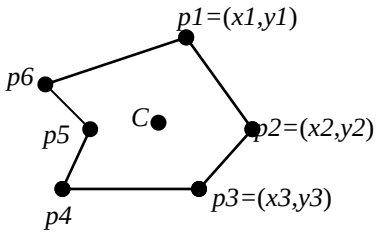

Figure 1: The representation of a polygon with six points or vertices (*i.e.*, *n=6*).

Given the declaration of class `Point` in the file **lab02.cpp**. Without changing any code of the class declaration, edit the file to accomplish the following task requirements:

1. Complete the definition of the overlaoded `operator +` (addition) which adds up each coordinate of two points. For example, if the two points are *p1* = (1,2) and *p2* = (5,7), then *p1* + *p2* = (6, 9).

2. Complete the definition of the overlaoded `operator -` (subtraction) which calculates the distance between two points. For example, if the two points are *p1* = (1,2) and *p2* = (5,7), then *p1* - *p2* = 6.403.

3. Complete the definition of the overlaoded `operator /` (division) which divides each coordinate with the right operand. For example, if the point is p = (1,2), then p / 2.0 = (0.5, 1.0).

4. Using the overloaded input operator and a proper loop, read a list of vertices from the keyboard.

5. Using another loop, calculate the center point by taking the average of the vertices of the polygon.

6. Using the overloaded output operator, print the center point onto the screen.

7. Using another loop, print the distance of each vertex to the center point of the polygon onto the screen.

Figure 2 and 3 show example runs of the program.

```
Enter the number of vertices of a polygon => 3
Enter vertex 1 => 0 0
Enter vertex 2 => 2.5 0
Enter vertex 3 => 3.5 9

The center is (2.0,3.0)

Vertex   Coordinates     Distance to center
1        (0.0,0.0)       3.4
2        (2.5,0.0)       3.0
3        (3.5,9.0)       6.1
```

Figure 2: Example runs of the program for a polygon with three vertices. User inputs are shown as bold texts

```
Enter the number of vertices of a polygon => 5
Enter vertex 1 => 1 1
Enter vertex 2 => 2 2
Enter vertex 3 => 0 4
Enter vertex 4 => 1.5 3
Enter vertex 5 => 0 0

The center is (0.9,2.0)

Vertex   Coordinates     Distance to center
1        (1.0,1.0)       1.0
2        (2.0,2.0)       1.2
3        (0.0,4.0)       2.2
4        (1.5,3.0)       1.0
5        (0.0,0.0)       2.2
```

Figure 3: Example runs of the program for a polygon with five vertices. User inputs are shown as bold texts