

**CS 1160-01 Introduction to Computer Science  
and Programming Methods  
Programming Assignment 6  
Due, Wednesday, March 9, 2005**

**A Short Wrap Up Combining Some Ideas  
Exploring Triangles**

The point of this exercise is to allow you to practice with some of the concepts we have encountered—structs, arrays, functions—in a fairly familiar environment in order to get a little more practice as to how these things work together. The basic task is to read in (as coordinates in the plane) three points to be the vertices of a *triangle* in the plane. Then you will compute its *centroid* (i.e., the location of its center of gravity if it were a uniform solid) and its *area*.

[*For the adventurous:* If you want to put your triangles into 3-space, i.e., have 3 coordinates for each point, *exactly* (well, with the “obvious” modifications) the same computations work.]

Start with a data type definition

```
struct point { float x; float y;};
```

Then you should define a function whose prototype is

```
float dist(point p, point q);
```

which will compute the *distance* between points *p* and *q*. [In case you have forgotten, the distance *d* between points  $(x, y)$  and  $(a, b)$  is given by  $d = \sqrt{(x - a)^2 + (y - b)^2}$ .]

You will also need a function to compute the *area* of the triangle. The easiest way to do this is to use Hero’s Formula, which gives the area of a triangle in terms of the *lengths* of the sides. Your function prototype should be something like

```
float triangleArea(float a, float b, float c);
```

where *a*, *b*, and *c* represent the *lengths* of the three sides. [Hero’s Formula says that the area of a triangle whose sides have lengths *a*, *b*, and *c* is given by

$$Area = \sqrt{s * (s - a) * (s - b) * (s - c)},$$

where  $s = (a + b + c)/2$ .]

You might also find a function something like `readPoint(point& pp)` useful for gathering the reading of points into one place. Maybe a `printPoint(point qq)` function would be useful, too, for printing out coordinates. You’ll have to decide what makes your code most easily understandable.

The *centroid* of a triangle is simply the point whose *x*-coordinate is the *average* of the *x*-coordinates of the vertices and whose *y*-coordinate is the average of the *y*-coordinates of the vertices. [If interested, one can prove that this point has the claimed center-of-gravity property either by using calculus or simply showing that every line through that point cuts the triangle into two equal areas.]

Your `main()` should define a variable “`point triangle[3];`” to hold the vertices of your triangle. This means, for example, that the vertex `triangle[0]` has *x*-coordinate `triangle[0].x` and *y*-coordinate `triangle[0].y`. You, of course, will prompt your user to provide coordinates for the three vertices, will *echo* what the vertices read were, and will do the two desired computations—centroid and area.

**What to turn in**

As usual, you should turn in your source code, which should be in good style and well commented. You should also turn in the results of enough runs of your program to convince one that it works correctly.