Section, Name and ID#:_

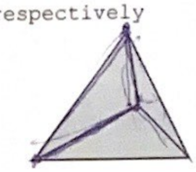
Problem 2: Implement a C++ *class triangle* (only its member functions marked by **TODO**) the header file of which is given below. The Heron's formula is $area = \sqrt{p(p-a)(p-b)(p-c)}$, where p is the half-perimeter and a, b and c are the sides.

OOP.809.140413.H057

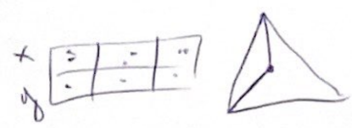
```
class triangle
{ public:
      triangle(double vertex[][3]); // TODO - initializes vertices by specified
                                    // array of two rows and three columns
      double get_x(int vertex); // returns x coordinate of specified vertex
      double get_y(int vertex); // returns y coordinate of specified vertex
      double side(int vertex); // returns side length from specified vertex to next one

      double perimneter(); // TODO
      double area(); // TODO - computes area using Heron's formula
      bool is_inside(double px, double py); // TODO - checks if a point  with coordinates
                          // (px, py) is inside the triangle - see shaded areas below
  private:
      double x[3], y[3]; // arrays of x and y coordinates of vertices respectively
}
```

3



```
#include <cmath>
triangle::triangle(double vertex[][3]) {
    x[0] = vertex[0][0];
    x[1] = vertex[0][1];
    x[2] = vertex[0][2];
    y[0] = vertex[1][0];
    y[1] = vertex[1][1];
    y[2] = vertex[1][2]; }
double triangle::perimeter() {
  double perimeter = dist(x[0], y[0], x[1], y[1]) + dist(x[1], y[1],
                     x[2], y[2]) + dist(x[2], y[2], x[0], y[0]);
  return perimeter;
       or
  double perimeter = side(0) + side(1) + side(2);
     return perimeter; }.
```



Student's copy

```
double triangle::area() { double p = perimeter()/2
   double area = sqrt(perimeter()/2 · (perimeter()/2 - side(0)) ·
   · (perimeter()/2 - side(1))(perimeter()/2 - side(2));
   double area = sqrt(p (p - side(0)) · (p - side(1)) · (p - side(2));
   return area; }
bool triangle::is_inside(double px, double py) {
   if (area(x[0], y[0], x[1], y[1], px, py) + area (x[1], y[1], x[2], y[2],
       , px, py) + area(x[2], y[2], x[0], y[0], px, py) = = area())
          return true; }
     else { return false; }. }
```

objects?

6 args.
area is a function that takes 3 points and finds the area of the triangle obtained by its

Use the backside, if needed


```
bool triangle::is_inside(double px, double py) {
  if (sqrt((dist(px, py) + side(0) + side(1))/2 (p1 - side(0) ·
      · (p1 - side(1) · (p1 - dist(px, py)) + 2 others (the same))
      return 1; }
  else { return 0; } };
```