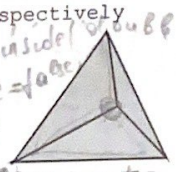
Section, Name and ID#:_

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.

```
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
}
```




```
triangle::triangle(double vertex[][3]){
    for(int i=0; i<3; i++){
        vertex[0][i]=x[i];
        vertex[1][i]=y[i];
    }
}
double triangle::perimeter(){
    double perimeter;
    perimeter = side(0)+side(1)+side(2);
    return perimeter;
}
double triangle::area(){
    double area;
    area = sqrt((perimeter()/2)*(perimeter()/2-side(0))
           *(perimeter()/2-side(1))*(perimeter()/2-side(2));
    return area;
}
```

(we already wrote them)

```
using(area,triangle
int area1, area3
int area2
int area S;
bool triangle::is_inside(double px, double py){
bool is_inside=false;
if(area S1+area S2+area S3 > area S){
    is_inside = true;
}
return is_inside;
}
```

Student's copy

OOP.Q09.140417.H164

Use the backside, if needed