

```

// Ploygon_PS1.cpp
// problem_Sets
// Created by H M Asfaq Ahmed Shihab on 24/3/2024.

#define _USE_MATH_DEFINES
#include <cassert>
#include <iostream>
#include "Polygon.hpp"
#include "Vector3D.hpp"

float Polygon::getSignedArea() const noexcept {
    if (fNumberOfVertices <= 2) {
        std::cerr << "Error: Polygon must have at least three vertices to
            calculate signed area.\n";
        return 0.0f;
    }

    float lArea = 0.0f;
    for (size_t i = 0; i < fNumberOfVertices - 1; i++) {
        lArea += 0.5f * (fVertices[i].y() + fVertices[i + 1].y()) *
            (fVertices[i].x() - fVertices[i + 1].x());
    }
    lArea += 0.5f * (fVertices[fNumberOfVertices - 1].y() +
        fVertices[0].y()) * (fVertices[fNumberOfVertices - 1].x() -
        fVertices[0].x());
    return lArea;
}

Polygon Polygon::transform(const Matrix3x3& aMatrix) const noexcept {
    Polygon lTransform = *this;
    Vector3D lTransformVec;
    for (size_t i = 0; i < fNumberOfVertices; i++) {
        lTransformVec = Vector3D(fVertices[i].x(), fVertices[i].y(), 1.0f);
        lTransformVec = aMatrix * lTransformVec;
        lTransform.fVertices[i] = Vector2D(lTransformVec.x(),
            lTransformVec.y());
    }
    return lTransform;
}

```