```cpp
//Polygon_PS1.cpp

// Created by NUR E SIAM

#include "Polygon.h"
#include <cmath>


float Polygon::getSignedArea() const noexcept {
    // Calculates the signed area of the polygon using the shoelace formula
    float larea = 0.0f;

    for (size_t lIndex = 0; lIndex < fNumberOfVertices; ++lIndex) {
        size_t j = (lIndex == fNumberOfVertices - 1) ? 0 : lIndex + 1; // Wrap around for closing edge

        larea += fVertices[lIndex].x() * fVertices[j].y() -
            fVertices[j].x() * fVertices[lIndex].y();
    }

    return larea / 2.0f;
}

Polygon Polygon::transform(const Matrix3x3& aMatrix) const noexcept {
    // Creates a new polygon by applying the given transformation matrix to each vertex
    Polygon ltransformed;
    ltransformed.fNumberOfVertices = fNumberOfVertices;

    size_t lIndex = 0;
    size_t rIndex = fNumberOfVertices - 1; // Start from opposite ends

    while (lIndex <= rIndex) {
        // Transform vertices from left to right
        Vector3D vertex = aMatrix * Vector3D(fVertices[lIndex].x(), fVertices[lIndex].y(), 1.0f);
        ltransformed.fVertices[lIndex] = Vector2D(vertex.x(), vertex.y());

        // Transform vertices from right to left (for potential efficiency)
        vertex = aMatrix * Vector3D(fVertices[rIndex].x(), fVertices[rIndex].y(), 1.0f);
        ltransformed.fVertices[rIndex] = Vector2D(vertex.x(), vertex.y());

        lIndex++;
        rIndex--;
    }

    return ltransformed;
}
```