

```

// COS30008, Tutorial 2, 2022

#include "Polygon.h"

#include

using namespace std;

Polygon::Polygon() : fNumberOfVertices(0) {}

size_t Polygon::getNumberOfVertices() const { return fNumberOfVertices; }

const Vector2D &Polygon::getVertex(size_t aIndex) const { if (aIndex < fNumberOfVertices) { return fVertices[aIndex]; }

throw out_of_range("Illegal index value."); }

void Polygon::readData(istream &aIStream) { // read input file containing 2D
vector data // if no data can be read, then exit loop // lInput » lVectors[lIndex]
evaluates to false on EOF while (aIStream » fVertices[fNumberOfVertices]) {
fNumberOfVertices++; } }

float Polygon::getPerimeter() const { float Result = 0.0f;

// There have to be at least three vertices if (fNumberOfVertices > 2) { //
solution without modulus and explicit temporary variables for (size_t i = 1; i <
fNumberOfVertices; i++) { Result += (fVertices[i] - fVertices[i - 1]).length(); }

Result += (fVertices[0] - fVertices[fNumberOfVertices - 1]).length();

}

return Result; }

Polygon Polygon::scale(float aScalar) const { Polygon Result = *this;

for (size_t i = 0; i < fNumberOfVertices; i++) { Result.fVertices[i] = fVertices[i]
* aScalar; }

return Result; }

```