ex3

Sheharyar Alam Khan

February 9, 2019

Contents

1 main.cpp

The Purpose

In this lab we want to find the perimeter of the triangle. the perimeter is the sum of all sides. a side is a line-segment. a line-segment is a line with a starting position and an ending position. a position on a cartesian plane can be represented by a point. A triangle consists of 3 points and its sides are the line segments between the points. We can use this to calculate the perimeter.The Proccess

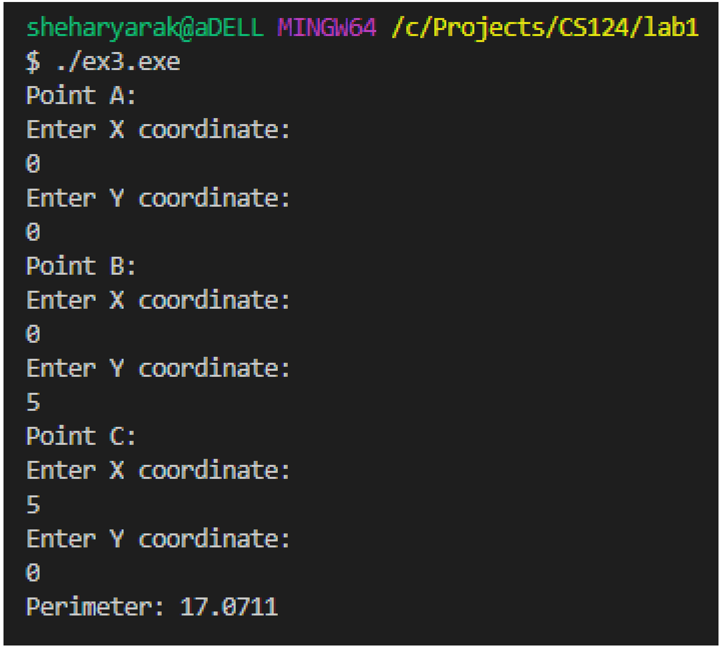
Since a triangle has three points and the line-segments between the points are the sides. And since the perimeter is the sum of of the length of all sides we can use our distance function to get the length of each of the sides and a perimeter function would just return their sum.

```
#include "point.hpp"
#include "triangle.hpp"

using namespace std;
```

This program gets three points using the getPoint() function which returns a point. it than creates a triangle structure using those points. Finally it prints out the perimeter onto the console using the perimeter function.

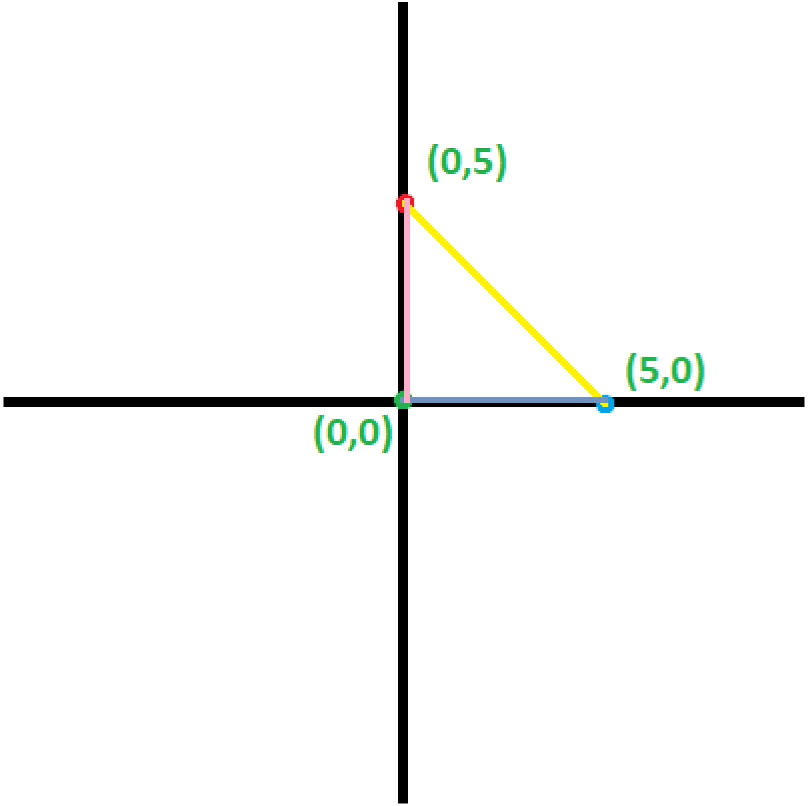
```
int main(void)
{
        cout << "Point A:" << endl;
        Point a = getPoint();

        cout << "Point B:" << endl;
        Point b = getPoint();

        cout << "Point C:" << endl;
        Point c = getPoint();
        Triangle tri(a, b, c);

        cout << "Perimeter: " << perimeter(tri) << endl;
}
```

This is what the output looks like:



```
sheharyarak@aDELL MINGW64 /c/Projects/CS124/lab1
$ ./ex3.exe
Point A:
Enter X coordinate:
0
Enter Y coordinate:
0
Point B:
Enter X coordinate:
0
Enter Y coordinate:
5
Point C:
Enter X coordinate:
5
Enter Y coordinate:
0
Perimeter: 17.0711
```

This is what the diagram looks like:



2 point.h

```
#ifndef POINT_H
#define POINT_H

#include <math.h>
#include <iostream>
using namespace std;

//previously explained in ex1.
struct  Point
{
        int x;
        int y;
        Point(void);
        Point(int xx, int yy);
};

Point   midpoint(Point a, Point b);
float   distance(Point a, Point b);
void    showCoords(Point a);
Point   getPoint(void);

#endif
```

3 point.cpp

```
#include "point.hpp"
#include <math.h>
#include <iostream>

//initializes the point to (0,0)
Point::Point()
{
        x = 0;
        y = 0;
}
//previously explained in ex1.
Point::Point(int xx , int yy)
{
        x = xx;
        y = yy;
}
//previously explained in ex1.
float distance(Point a, Point b)
{
        return (sqrt(((b.y- a.y) * (b.y - a.y)) + ((b.x - a.x) * (b.x - a.x)
            )));
}
//previously explained in ex2.
Point midpoint(Point a, Point b)
{
        int x;
        int y;

        x = (a.x + b.x) / 2;
        y = (a.y + b.y) / 2;
        Point mid(x,y);
        return (mid);
}
//previously explained in ex2.
void    showCoords(Point a)
{
        cout << "X: " << a.x <<endl;
        cout << "Y: " << a.y <<endl;
        cout << endl;
}
//previously explained in ex1.
Point getPoint(void)
{
        int x;
        int y;

        cout << "Enter X coordinate:" << endl;
        cin >> x;
        cout << "Enter Y coordinate:" << endl;
        cin >> y;
        Point p(x,y);
        return (p);
}
```

4 triangle.h

```
#ifndef TRIANGLE_H
#define TRIANGLE_H

#include "point.hpp"
```

`struct Triangle`

This structure consists of three points. It also has two Triangle constructors. A default constructor that just makes a Triangle and a Constructor that takes in three Points.

```
struct  Triangle
{
        Point a;
        Point b;
        Point c;
        Triangle(void);
        Triangle(Point aa, Point bb, Point cc);
};

float   perimeter(Triangle tri);

#endif
```

5 triangle.cpp

```
#include "triangle.hpp"
#include "point.hpp"
```

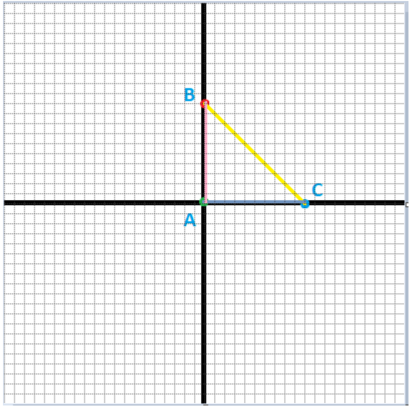
Triangle::Triangle(Point aa, Point bb, Point cc)

This constructor takes in three points, aa, bb, and cc, and sets them equal to a, b, and c.

```
Triangle::Triangle(Point aa, Point bb, Point cc)
{
        a = aa;
        b = bb;
        c = cc;
}
```

`float perimeter(Triangle tri)`

This function uses the distance function to calculate the distance between points A and B, B and C, and C and A it then returns the sum of these distances. This sum is the perimeter of the triangle.



```
float   perimeter(Triangle tri)
{
        float ab;
        float bc;
        float ca;

        ab = distance(tri.a, tri.b);
        bc = distance(tri.b, tri.c);
        ca = distance(tri.c, tri.a);
        return (ab + bc + ca);
}
```