ex1

Sheharyar Alam Khan

February 9, 2019

Contents

1 main.cpp

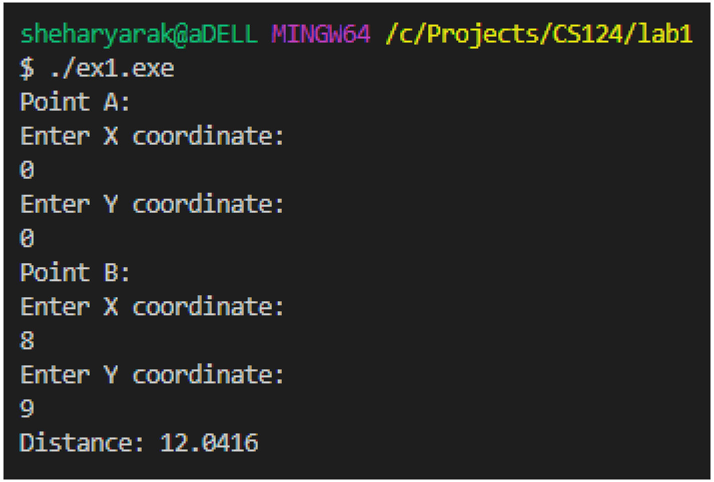
The Purpose
The purpose of this exercise is to revise the concepts of C++ structures. Structure are a core part of data storage in C++ and it is crucial that we know how to use them. In this exercise we want to use a structure that holds an x coordinate and a y coordinate and calculate the distance between them.
The Proccess
To do this we will use the distance formula derived by mathematicians years ago. In this program we're using two functions, a structure, and a constructor. These will be explained below.

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

using namespace std;
```

The main function gets the points from the user using the *getPoint()* function and prints out the distance to the console using the *distance(Point a, Point b)* function.

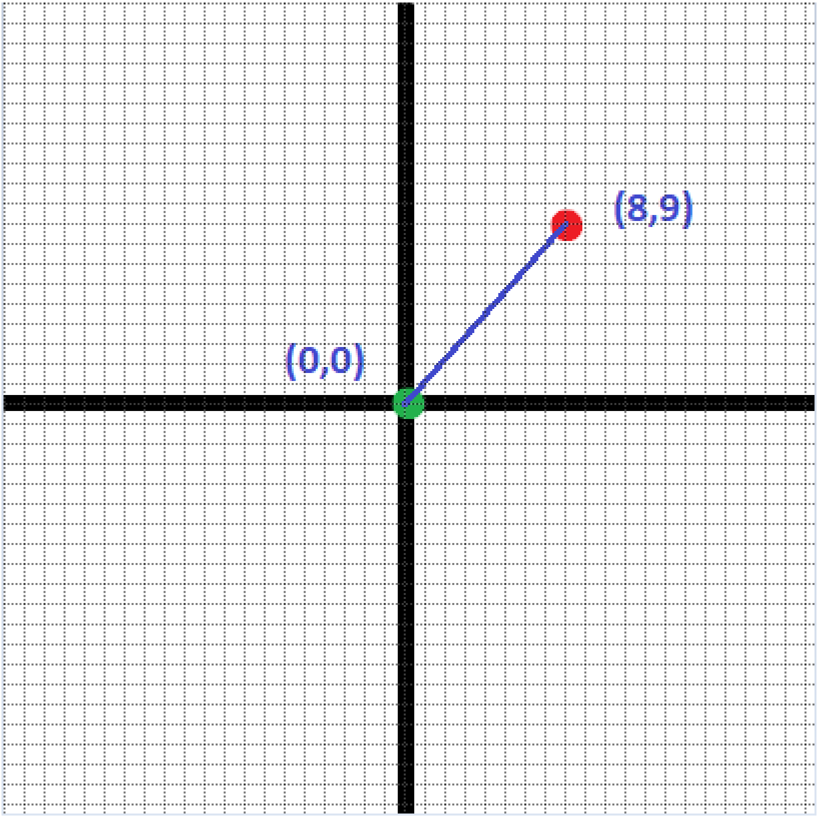
```
int main(void)
{
        cout << "Point A:" << endl;
        Point a = getPoint();

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

        cout << "Distance: " << distance(a,b) << endl;
        return (0);
}
```

This is what the output looks like:



```
sheharyarak@aDELL MINGW64 /c/Projects/CS124/lab1
$ ./ex1.exe
Point A:
Enter X coordinate:
0
Enter Y coordinate:
0
Point B:
Enter X coordinate:
8
Enter Y coordinate:
9
Distance: 12.0416
```

```
This is what the diagram looks like:
```



2 point.h

```
#ifndef POINT_H
#define POINT_H

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

struct Point

A Point, by its mathematical, definition is a location on a plane. In our case, this is a 2D cartesian plane. This implies that in order to describe a location on a plane our Point must have and X displacement from the orgin and a Y displacement from the origin.Since our Point needs to values, an X-displacement and a Y-displacement, our structure contains two floats x and y. it also contains a constructor.The constructor takes in two floats, xx and yy, and sets them equal to x and y which makes it easier (in my opinion) to define the Point. We're using floats because we want decimals.

```
struct  Point
{
        float x;
        float y;
        Point(int xx, int yy);
};

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

#endif
```

3 point.cpp

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

using namespace std;
```

`Point(float xx, float yy)`

Point(float xx, float yy) is a customized constructor for the structure Point.
It takes in the X coordinate and the Y coordinates as arguments and sets them equal to X and Y accordingly.

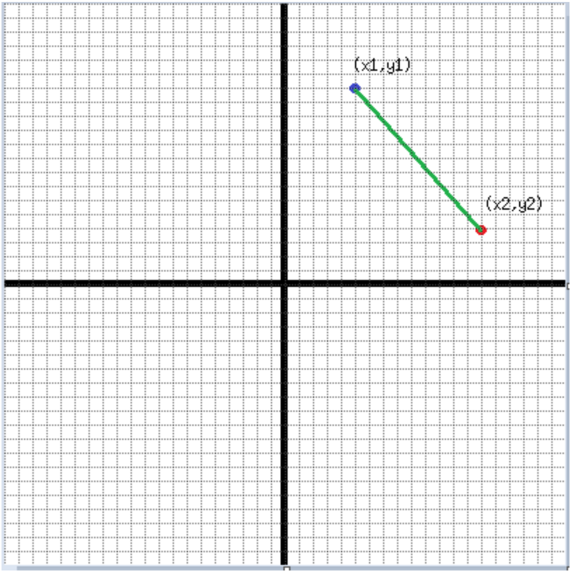
```
Point::Point(float xx , float yy)
{
        x = xx;
        y = yy;
}
```

```
float distance(Point a, Point b)
```

This function implements the distance formula and returns the result.
The distance formula is shown below:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

This diagram illustrates the distance formula:



```
float distance(Point a, Point b)
{
        return (sqrt(((b.y- a.y) * (b.y - a.y)) + ((b.x - a.x) * (b.x - a.x)
            )));
}
```

Point getPoint(void)

This function takes input from the user.It then creates a point using the Point constructor and the values provided and returns the Point.

```
Point getPoint(void)
{
        float x;
        float y;

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