ex2

Sheharyar Alam Khan

February 9, 2019

Contents

1 main.cpp

The Purpose
    The Purpose of this exercise and the entire lab is to review basic C++ concepts before moving forward with the semester. In this exercise we are required to find a point that lies directly in between two points.
    The Process
    Exactly like in ex1, we will first get two points from the user and then display the midpoint onto the console using the midpoint function.

```
#include "point.hpp"

using namespace std;
```

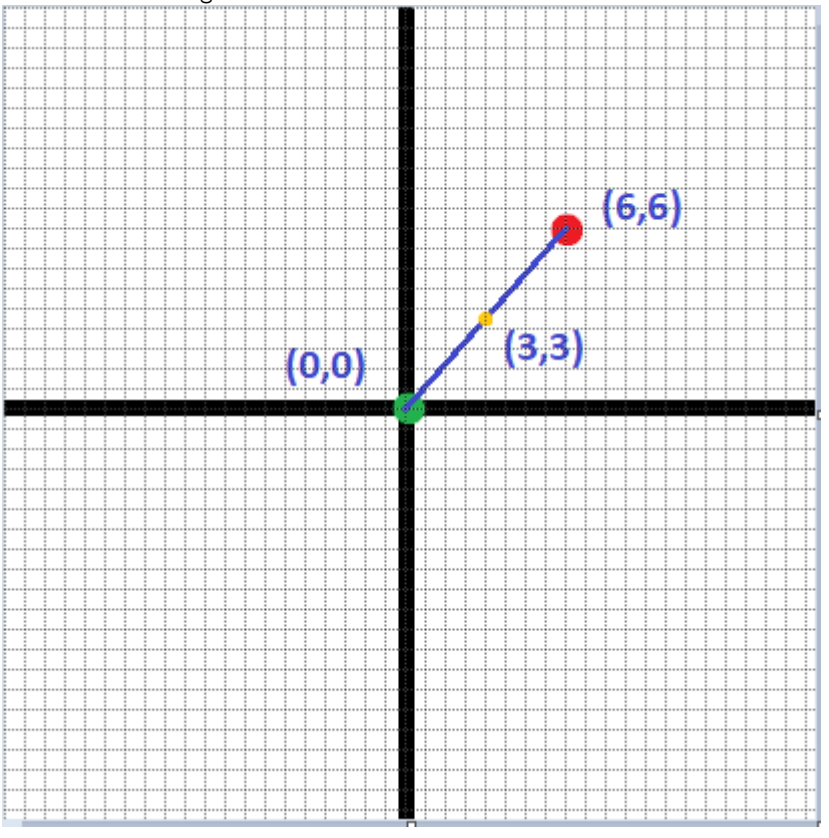
The main function gets the points from the user and prints out the coordinates of the midpoint to the console.

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

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

        cout << "Midpoint Coords: " << endl;
        showCoords(midpoint(a,b));

        return (0);
}
```

This is what the output looks like:

```
sheharyarak@aDELL MINGW64 /c/Projects/CS124/lab1
$ ./ex2.exe
Point A:
Enter X coordinate:
0
Enter Y coordinate:
0
Point B:
Enter X coordinate:
6
Enter Y coordinate:
6
Midpoint Coords:
X: 3
Y: 3
```

This is what the diagram looks like:



2 point.h

```
#ifndef POINT_H
#define POINT_H

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

This structure was previously explained in ex1.
struct  Point
{
        float x;
        float y;
        Point(float xx, float yy);
};

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

#endif
```

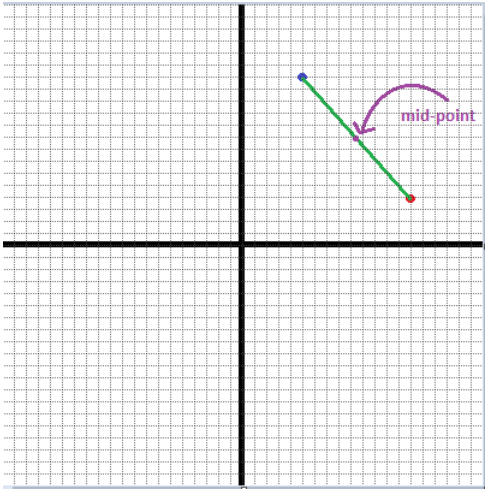
3 point.cpp

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


This function was previously explained in exercise 1.
Point::Point(float xx , float yy)
{
        x = xx;
        y = yy;
}
```

`Point midpoint(Point a, Point b)`

This function uses the midpoint formula to calculate the midpoint.

$$\left(\frac{x_2 - x_2}{2}, \frac{y_2 - y_1}{2}\right)$$




```
Point midpoint(Point a, Point b)
{
        float x;
        float y;

        x = (a.x + b.x) / 2;
        y = (a.y + b.y) / 2;
        Point mid(x,y);
        return (mid);
}
```

`void showCoords(Point a)`

This function takes in a Point and prints out its x and y values on to the console.

```
void     showCoords(Point a)
{
         cout << "X: " << a.x <<endl;
         cout << "Y: " << a.y <<endl;
         cout << endl;
}
```

This function was previously explained in ex1.

```
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);
}
```