Handed out: June-24.

SWE315 : C++

Homework 5 (there was NO 4) - (Due date 07-01 : a Week!!)

Solution:

1. Please send solution to: zbaharav@cogswell.edu
2. You know the drill by now: Simply hit reply, and no zipped directories etc.. Just ascii-files or Word documents (or equivalent)

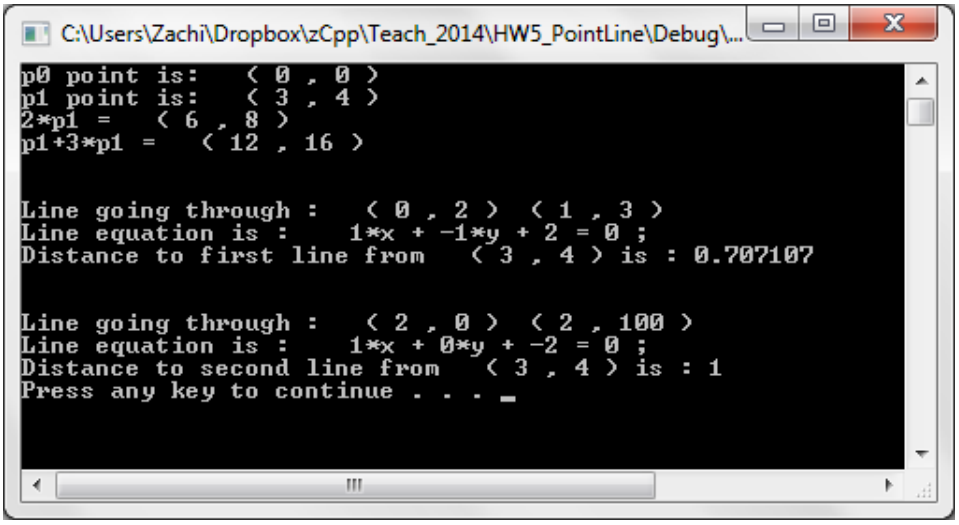
====

In this homework you will create (2D!) Point and Line classes.

Please read the below TO-THE-END, as there are important hints at the very bottom.

a. The Point class should have the following interfaces:
    i. Constructor:
        1. Empty (set point to origin (0,0) .
        2. Given (x,y)
    ii. Two getter functions:
        1. getx() function.
        2. gety() function.
    iii. Two distance functions:
        1. Distance from another point.
        2. Distance from a line.
    iv. Overloaded operators:
        1. Plus '+'
        2. Multiple by a float '*'
        3. Output '<<'
b. The Line class should have the following interfaces:
    i. Constructor:
        1. Given two points, define a line.
    ii. One distance functions:
        1. Distance from a point.
    iii. Overloaded operators:
        1. Output '<<'

This is how the output of main.cpp (testing program) should look like:



And this is the main.cpp program itself:

```
// File: main.cpp

#include <iostream>

#include "Point.h"
#include "Line.h"

using std::cout;
using std::endl;

int main(void)
{

	Point p0;
	cout << "p0 point is: " << p0 << endl;

	Point p1(3,4);
	cout << "p1 point is: " << p1 << endl;

	cout << "2*p1 = " << 2*p1 << endl;
	cout << "p1+3*p1 = " << p1+3*p1 << endl;


	Point pline1(0,2), pline2(1,3);
	Line line1(pline1,pline2);
	cout << "\n\nLine going through : " << pline1 << pline2 << endl;
	cout << "Line equation is : "  << line1 << endl;
	cout << "Distance to first line from " << p1 << " is : "  <<
p1.distance(line1) << endl;

	Point pline3(2,0), pline4(2,100);
	Line line2(pline3,pline4);
	cout << "\n\nLine going through : " << pline3 << pline4 << endl;
	cout << "Line equation is : "  << line2 << endl;
	cout << "Distance to second line from " << p1 << " is : "  <<
line2.distance(p1) << endl;
```

```
        system("pause");

        return 0;
}
```

Hints (or guidelines):

1. Represent the line using 3 parameters (of course only two are needed, but this will keep things simple).
   ```
   // We will use representation:
   // ax+by+c = 0
   ```
1. When initializing a line, pay attention to the case of vertical line!
2. For a non-vertical line, the values of a,b,c above are given by:

   a = y2 – y1

   b = x1 – x2

   c = y1*x2 – x1*y2

   CHECK these on a piece of paper before proceeding to programming: It will save you a lot of heartache!

3. Distance from a point to a line:
   http://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line

$$\operatorname{distance}(ax+by+c=0,(x_0,y_0)) = \frac{|ax_0+by_0+c|}{\sqrt{a^2+b^2}}.$$

=== End of Homework 5 ====