CECS 524 Unit 11 Assignment

Name:Spuritha Mudireddy
CSULB ID:030743269

point.h

```
#ifndef POINT_H
#define POINT_H
class Point {
double x, y;
public:
double getX() ;
double getY() ;
void setX(double);
void setY(double);
};
#endif
```

line.h

```
#ifndef LINE_H
#define LINE_H
#include "point.h"
#include <iostream>

using namespace std;

class line {

Point p1, p2;

public:

line();

line(double, double, double, double);

line(line& l);

void SetPoint1(double, double);
```

```
void SetPoint2(double, double);
void SetLine(double, double, double, double);
bool operator==(line& l);
bool operator!=(line& l);
friend ostream& operator<<(ostream &out, line& l);
friend istream& operator>>(istream &in, line &l);
double Distance();
double Slope();
};
#endif
```

point.cpp

```
#include "point.h"
double Point::getX() {
return x;
}

double Point::getY()  {
return y;
}
void Point::setX(double x) {
this->x = x;
}
```

```
void Point::setY(double y) {
this->y = y;
}
```

line.cpp

```
#include "line.h"
#include <cmath>

line::line() {}

line::line(double x1, double y1, double x2, double y2) {
p1.setX(x1);
p1.setY(y1);
p2.setX(x2);
p2.setY(y2);
}

line::line(line& l) {
p1.setX(l.p1.getX());
p1.setY(l.p1.getY());
p2.setX(l.p2.getX());
p2.setY(l.p2.getY());
}
```

```
void line::SetPoint1(double x1, double y1) {
p1.setX(x1);
p1.setY(y1);
}

void line::SetPoint2(double x2, double y2) {
p2.setX(x2);
p2.setY(y2);
}

void line::SetLine(double x1, double y1, double x2, double y2) {
p1.setX(x1);
p1.setY(y1);
p2.setX(x2);
p2.setY(y2);
}


bool line::operator==(line& l) {
if ((p1.getX() == l.p1.getX()) &&
(p1.getY() == l.p1.getY()) &&
(p2.getX() == l.p2.getX()) &&
(p2.getY() == l.p2.getY()))
```

```
return true;


return false;

}


bool line::operator!=(line& l) {


if ((p1.getX() != l.p1.getX()) ||

(p1.getY() != l.p1.getY()) ||

(p2.getX() != l.p2.getX()) ||

(p2.getY() != l.p2.getY()))

return true;

return false;

}

 ostream& operator<<(ostream &out, line& l) {

out << "(" << "(" << l.p1.getX() << "," << l.p1.getY() << ")" << "(" << l.p2.getX() << ","
<< l.p2.getY() << ")" << ")";

return out;

}


 istream& operator>>(istream &in, line &l) {

double x1, y1, x2, y2;

in >> x1 >> y1 >> x2 >> y2;

l.p1.setX(x1);

l.p1.setY(y1);
```

```
l.p2.setX(x2);

l.p2.setY(y2);

return in;

}

double line::Distance() {

double d = sqrt(pow(p2.getX() - p1.getX(), 2) + pow(p2.getY() - p1.getY(), 2));

return d;

}


double line::Slope() {

double s = (p2.getY() - p1.getY()) / (p2.getX() - p1.getX());

return s;

}
```

main.cpp

```
#include "line.h"

int main() {
line a;
a.SetPoint1(2, 8);
a.SetPoint2(5, 11);
line b(2, 8, 5, 11);
line c(b);
line d;
d.SetLine(9,6,4,2);
cout << "line a: " <<a<< endl;

cout << "line b: " << b<< endl;

cout << "line c: " << c << endl;
cout << "line d: " << d << endl;
```

```
if (a == b) {

cout << "Line a equals Line b"<<endl;

}

if (c != d) {

cout << "Line c not equals Line d"<<endl;

}
line e;

cout << "Enter line e: ";

cin >> e;

cout << "line e: " << e << endl;


cout << "Length of e: " << e.Distance() << endl;

cout << "Slope of e: " << e.Slope() << endl;


}
```

Output:

```
C:\Users\mspur\OneDrive\Desktop\Assignments\APL\Unit 11\code>g++ point.cpp line.cpp main.cpp -o a.exe

C:\Users\mspur\OneDrive\Desktop\Assignments\APL\Unit 11\code>a.exe
line a: ((2,8)(5,11))
line b: ((2,8)(5,11))
line c: ((2,8)(5,11))
line d: ((9,6)(4,2))
Line a equals Line b
Line c not equals Line d
Enter line e: 1 2 3 4
line e: ((1,2)(3,4))
Length of e: 2.82843
Slope of e: 1
```