1. **main.cpp**

#include "line.h"

int main() {

Line l1(2, 5, 8, 7);

Line l2(l1);

Line l3;

l3.SetPoint1(10, 12);

l3.SetPoint2(7, 5);

cout << "Coordinates of Line l1: " << l1 << endl;

cout << "Coordinates of Line l2: " << l2 << endl;

cout << "Coordinates of Line l3: " << l3 << endl;

Line l4;

cout << "Enter Line L4: ";

cin >> l4;

cout << "Coordinates of Line L4: " << l4 << endl;

cout << "Length of L1: " << l1.Distance() << endl;

cout << "Slope of L1: " << l1.Slope() << endl;

if (l1 == l2)

cout << "L1 is equal to L2\n";

if (l3 != l4)

cout << "L3 is not equal to L4\n";

}

1. **line.h**

#ifndef LINE\_H

#define LINE\_H

#include "point.h"

#include <iostream>

using namespace std;

class Line {

Point point1, point2;

public:

Line();

Line(double x1, double y1, double x2, double y2);

Line(Line& I);

void SetPoint1(double x1, double y1);

void SetPoint2(double x2, double y2);

void SetLine(double x1, double y1, double x2, double y2);

double Distance();

double Slope();

friend ostream& operator <<(ostream &output, const Line& line);

friend istream& operator >>(istream &input, Line &line);

bool operator==(const Line& line);

bool operator!=(const Line& line);

};

#endif

1. **line.cpp**

#include "line.h"

#include <cmath>

Line::Line() {}

Line::Line(double x1, double y1, double x2, double y2) {

point1.setX(x1);

point1.setY(y1);

point2.setX(x2);

point2.setY(y2);

}

Line::Line(Line& line) {

point1.setX(line.point1.getX());

point1.setY(line.point1.getY());

point2.setX(line.point2.getX());

point2.setY(line.point2.getY());

}

void Line::SetPoint1(double x1, double y1){

point1.setX(x1);

point1.setY(y1);

}

void Line::SetPoint2(double x2, double y2) {

point2.setX(x2);

point2.setY(y2);

}

void Line::SetLine(double x1, double y1, double x2, double y2) {

point1.setX(x1);

point1.setY(y1);

point2.setX(x2);

point2.setY(y2);

}

double Line::Distance(){

double distance = sqrt(pow(point2.getX() - point1.getX(), 2) + pow(point2.getY() - point1.getY(), 2));

return distance;

}

double Line::Slope() {

double slope = (point2.getY() - point1.getY()) / (point2.getX() - point1.getX());

return slope;

}

bool Line::operator==(const Line& line) {

if ((point1.getX() == line.point1.getX()) &&

(point1.getY() == line.point1.getY()) &&

(point2.getX() == line.point2.getX()) &&

(point2.getY() == line.point2.getY()))

return true;

return false;

}

bool Line::operator!=(const Line& line) {

if ((point1.getX() != line.point1.getX()) ||

(point1.getY() != line.point1.getY()) ||

(point2.getX() != line.point2.getX()) ||

(point2.getY() != line.point2.getY()))

return true;

return false;

}

ostream& operator <<(ostream &output, const Line& line) {

output << "(" << "(" << line.point1.getX() << "," << line.point1.getY() << ")" << "(" << line.point2.getX() << "," <<

line.point2.getY() << ")" << ")";

return output;

}

istream& operator >>(istream &input, Line &line) {

double x1, y1, x2, y2;

input >> x1 >> y1 >> x2 >> y2;

line.point1.setX(x1);

line.point1.setY(y1);

line.point2.setX(x2);

line.point2.setY(y2);

return input;

}

1. **point.h**

#ifndef POINT\_H

#define POINT\_H

class Point{

double x, y;

public:

void setX(double X);

void setY(double Y);

double getX() const;

double getY() const;

};

#endif

1. **point.cpp**

#include "point.h"

void Point::setX(double X){

x = X;

}

void Point::setY(double Y){

y = Y;

}

double Point::getX() const{

return x;

}

double Point::getY() const{

return y;

}

**Output**

