Answer



* Note there are 5 files:
* 1. point.h
* 2. point.cpp
* 3. line.h
* 4. line.cpp
* 5. main.cpp
* use the command to compile the code:

\* g+ + main.cpp line.cpp point.cpp

//point.h

#ifndef POINT\_H #define POINT\_H

//class point

class Point (

double x, y; //coordinates variables

public:

void setX(double); //sets x coordinate

void setY(double); //sets y coordinate

double getX() const; //return x coordinate double getY() const; //return y coordinate

#endif

//P oi»t-P PP

#incIude "point.h"

//sets x coordinate

void Point::setX(double x) ( this->x = x;

//sets y coordinate

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

//return x coordinate double Point::getX() const (

return x;

//return y coordinate

double Point::getY() const ( return y;

//line.h

#ifndef LINE\_H #define LINE\_H #incIude "point.h" #incIude < iostream> using namespace stdi

class line (

Point point1, point2; //staring point and ending point public:

line(); //default constructor

line(double, double, double, double); //parametrized constructor line(Iine& I); //copy constructor

void SetPoint1(double, double); //sets starting point void SetPoint2(double, double); //sets ending point

void SetLine(double, double, double, double); //sets both the points double Distance(); //retruns length of the line

double Slope(); //returns slope of the line

friend ostream& operators <(ostream &out, const Iine& I); //output line to the console friend istream& operator> >(istream &in, line &I); //input line from the console

bool operator==(const Iine& I); //returns true if two line are equal bool operator!=(const Iine& I); //returns true if two lines are not equal

#endif

//line.cpp

#incIude "line.h" #incIude < cmath>

//default constructor line::Iine() {)

//parametrized constructor

line::Iine(double x1, double y1, double x2, double y2) { point1.setX(x1);

point1.setY(y1); point2.setX(x2); point2.setY(y2);

//copy constructor line::Iine(Iine& I) (

point1.setX(I.point1.getX()); point1.setY(I.point1.getY()); point2.setX(I.point2.getX()); point2.setY(I.point2.getY());

//sets starting point

void line::SetPoint1(double x1, double y1) ( point1.setX(x1);

point1.setY(y1);

//sets ending point

void line::SetPoint2(double x2, double y2) { point2.setX(x2);

point2.setY(y2);

//sets both the points

void line::SetLine(double x1, double y1, double x2, double y2) ( point1.setX(x1);

point1.setY(y1); point2.setX(x2); point2.setY(y2);

//retruns length of the line double line::Distance() (

//using distance formula to calculate the lenpth

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

return distance;

//returns slope of the line

double line::SIope() (

//using the formual: y2-y1 / x2-x1 to calculate the slope

double slope = (point2.getY0 - point1.getY()) / (point2.getX() - point1.getX());

return slope;

//returns true if two line are equal bool line::operator==(const Iine& I) (

//checking all the corresponding coordinates in both the lines

//all the corresponding coordinates need to be equal

if ((point1.getX() == I.point1.getX()) && (point1.getY() == I.point1.getY()) && (point2.getX() == I.point2.petX()) && (point2.getY() == I.point2.getY())) return true;

return false;

//returns true if two lines are not equal

bool line::operator!=(const Iine& I) (

//checking all the corresponding coordinates in both the lines

//even if a single coordinate is found unequal it returns true

if ((point1.getX() != I.point1.getX()) || (point1.getY() != I.point1.getY()) || (point2.getX() != I.point2.getX()) || (point2.getY() != I.point2.getY())) return true;

return false;

//output line to the console

ostream& operators <(ostream &out, const Iine& I) {

out < < "(" < < "(" < < I.point1.getX() < < "," < < I.point1.getY() < < ")" < < "

(" < < I.point2.getX() < < "," < < I.point2.getY() < < ")" < < ")"; return out;

//input line from the console

istream& operator> >(istream &in, line &I) (

double x1, y1, x2, y2;

in > > x1 > > y1 > > x2 > > y2; I.point1.setX(x1); I.point1.setY(y1); I.point2.setX(x2); I.point2.setY(y2);

return in;

//main.cpp #incIude "line.h"

int main() (

line I1(1, 2, 3, 4); //using constructor line 12(I1); //using copy constructor

//settting points using methods line 13;

13.SetPoint1(5, 6);

13.SetPoint2(9, 10);

//printing lines

cout < < "line I1: " < < I1 < < endl; cout < < "line 12: " < < 12 < < endl; cout < < "line 13: " < < 13 < < endl;

//reading line from the user line 14;

cout < < "Enter line L4: "; cin > > 14;

cout < < "line L4: " < < 14 < < endl;

//calling Distance() and Slope() method

cout < < "length of L1: " < < I1.Distance() < < endI; cout < < "slope of L1: " < < I1.Slope() < < endl;

//Demonstrating comparison operators if (11 == 12) {

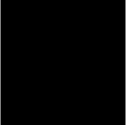
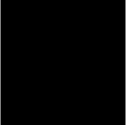
cout < < "yes L1 is equal to L2\n";

if (12 != 13) (

cout < < "L2 is not equal to L3\n";



View image!

Likes: 0 Dislikes: 0