```cpp
/*  Program: A3P2 - Pointland!
    Author: Tom Stutler
    Last Date Modified: 3/19/15

    The intent of this program is to repeatedly provide the user with a menu of choices to calculate/simulate
    a variety of tasks on an xy-coordinate plane given the user's entered points (x,y).
*/

#include <iostream>
#include <cmath>
using namespace std;

///Define new type point which stores the x and y coordinates of a point on 2D plane.
struct point
{
     float x;
     float y;
};

float dist (point aParam, point bParam);
///The purpose of this function is to take two points and return the distance between them on the xy-coordinate plane.

float slope (point aParam, point bParam, bool& defParam);
///The purpose of this function is to take two points and return the slope of the line connecting them.
///If there is no slope defined the return NULL and set slopeDefined as false.

point midpoint (point aParam, point bParam);
///The purpose of this function is to take two points and return the point that is in the middle
///of the line segment connecting them.

void equation (point aParam, point bParam);
///The purpose of this function is to take two point calculate and display the equation for the line
///that connects the two points together.

bool collinear (point aParam, point bParam, point cParam);
///The purpose of this function is to take three points and return true if they are collinear or false if not.

point readpt ();
///The purpose of this function is to prompt the user to enter a point in standard format (x,y)
///and return a point variable.

void showpt (point aParam);
///The purpose of this function is to take a point and display it to the user in standard format (x,y).

int main()
{
    //Initialize loop variable.
    bool repeat=true;

    //Loop to repeat until user selects to exit.
    do {
        //Declare variables.
        point a, b, c, mid;
        bool slopeDefined, isCollinear;
        float m;
        int selection;

        //Display menu to user.
        cout << "POINTLAND\n"
             << "What do you want to do?\n"
             << "1 - Find the distance between two points\n"
             << "2 - Find slope\n"
             << "3 - Find a midpoint\n"
             << "4 - Find an equation of a line\n"
             << "5 - Determine if three points are collinear\n"
             << "6 - Exit\n";


        //Prompt user for menu selection.
        cout << "Selection => ";
        cin >> selection;

        //Switch branch to carry out user's selection.
        switch (selection)
        {
        default:
            cout << "Please enter a valid input\n";
            break;

        case 1:
            //Prompt user for two points, pass them through dist(), then display the distance.
            cout << "Enter point 1: ";
            a = readpt();
            cout << "Enter point 2: ";
            b = readpt();
            cout << "Distance = " << dist(a, b) << endl;
            break;

        case 2:
            //Prompt user for two points, pass them through slope(), then display the slope.
            cout << "Enter point 1: ";
            a = readpt();
            cout << "Enter point 2: ";
            b = readpt();

            m = slope(a, b, slopeDefined);

            if (slopeDefined == true) {
                cout << "Slope = " << m << endl;
            } else {
                cout << "Slope is not defined for a vertical line.\n";
            }
            break;

        case 3:
            //Prompt user for two points, pass them through midpoint(), then display the midpoint.
            cout << "Enter point 1: ";
            a = readpt();
            cout << "Enter point 2: ";
            b = readpt();

            mid = midpoint(a, b);

            cout << "Midpoint = ";
            showpt(mid);
            break;

        case 4:
            //Prompt user for two points, pass them through equation().
            cout << "Enter point 1: ";
            a = readpt();
            cout << "Enter point 2: ";
            b = readpt();

            equation(a, b);
            break;

        case 5:
            //Prompt user for three points, pass them through collinear(), then display if the points are collinear.
            cout << "Enter point 1: ";
            a = readpt();
            cout << "Enter point 2: ";
            b = readpt();
            cout << "Enter point 3: ";
            c = readpt();

            isCollinear = collinear(a, b, c);

            if (isCollinear == true) {
                cout << "These points are collinear.\n";
            } else {
                cout << "These points are not collinear.\n";
            }
            break;

        case 6:
            cout << "Thanks for using POINTLAND\n";
            repeat = false;
            break;
        }

        //Display blank line between runs.
        cout << "\n";
    } while (repeat == true);

}

float dist (point aParam, point bParam)
{
    return sqrt(pow((aParam.x - bParam.x), 2) + pow((aParam.y - bParam.y), 2));
}

float slope (point aParam, point bParam, bool& defParam)
{
    if (aParam.x == bParam.x) {
        defParam = false;
        return 999;
    } else {
        defParam = true;
        return ((bParam.y - aParam.y)/(bParam.x - aParam.x));
    }
}

point midpoint (point aParam, point bParam)
{
    point middle;

    middle.x = (aParam.x + bParam.x)/2;
    middle.y = (aParam.y + bParam.y)/2;

    return middle;
}

void equation (point aParam, point bParam)
```

```
{
    float m, b;
    bool slopeDefined;

    m = slope(aParam, bParam, slopeDefined);
    b = (aParam.y - m * aParam.x);

    if ((slopeDefined == true) && (m != 0)) {
        cout << "Equation: y = " << m << "x+" << b << endl;
    } else if ((slopeDefined == true) && (m == 0)) {
        cout << "Equation: y = " << b << endl;
    } else {
        cout << "Equation: x = " << aParam.x << endl;
    }
}

bool collinear (point aParam, point bParam, point cParam)
{
    float m1, m2, m3;
    bool slopeDefined;

    m1 = slope(aParam, bParam, slopeDefined);
    m2 = slope(aParam, cParam, slopeDefined);
    m3 = slope(bParam, cParam, slopeDefined);

    if ((m1 == m2) || (m1 == m3) || (m2 == m3)) {
        return true;
    } else {
        return false;
    }
}

point readpt ()
{
    point userPoint;
    char parenL, parenR, comma;

    cin >> parenL >> userPoint.x >> comma >> userPoint.y >> parenR;

    return userPoint;
}

void showpt (point aParam)
{
    cout << "(" << aParam.x << "," << aParam.y << ")\n";
}
```