PL Assignment 11:

Output:

```
(base) darshanuttammistry@Darshans-MacBook-Air PL Assignment 11 % ./main
line 1: ((10,3)(11,3))
line 2 : ((10,3)(11,3))
Distance : 1
Slope : 0
if both points are Equal (1 for true, 0 for False) : 0
Enter X and Y co-ordinates for Point 1 and Point 2
2 4 2 4
New Line : ((2,4)(2,4))
```

Source code:

Point1.h

```
class point1
{
private:
    double x;
    double y;

public:
    point1();
    point1(point1 &p1);
    point1(double x1, double y1);

    double getpointX();
    double getpointY();
    void setpointX(double PX);
    void setpointY(double PY);
};
```

Point1.cpp

```
#ifndef Point1_H
#define Point1_H
#include "point1.h"
```

```
point1::point1() {}

point1::point1(point1 &p1)
{
    this->x = p1.x;
    this->y = p1.y;
}
point1::point1(double x1, double y1)
{
    this->x = x1;
    this->y = y1;
}
double point1::getpointX()
{
    return this->x;
}
double point1::getpointY()
{
    return this->y;
}
void point1::setpointX(double PX)
{
    this->x = PX;
}
void point1::setpointY(double PY)
{
    this->y = PY;
}
#endif
```

Line1.h

```
#include "point1.cpp"
class line1
{
private:
    point1 point11;
```

```
    point1 point2;
    /* data */
public:
    line1(/* args */);
    line1(point1 p1, point1 p2);
    line1(line1 &l);

    void setPoint1(double x, double y);

    void setPoint2(double x, double y);

    double Distance();

    double Slope();

    void setline1(double x1, double y1, double x2, double y2);

    point1 getPoint1();

    point1 getPoint2();

    bool equalTo();
};
```

Line1.cpp

```
#include <iostream>
#include "line1.h"
#include <cmath>
using namespace std;
line1::line1(point1 p1, point1 p2)
{
    this->point11 = p1;
    this->point2 = p2;
}
line1::line1(line1 &l)
{
```

```
    this->point11 = l.point11;
    this->point2 = l.point2;
}

void line1::setPoint1(double x, double y)
{
    this->point11.setpointX(x);
    this->point11.setpointY(y);
}

void line1::setPoint2(double x, double y)
{
    this->point2.setpointX(x);
    this->point2.setpointY(y);
}

double line1::Distance()
{
    double distance;

    distance = pow(pow(this->point2.getpointX() - this-
>point11.getpointX(), 2) + pow(this->point2.getpointY() - this-
>point11.getpointY(), 2), 0.5);

    return distance;
}

double line1::Slope()
{
    double slope;

    slope = (this->point2.getpointY() - this->point11.getpointY())
/ (this->point2.getpointX() - this->point11.getpointX());

    return slope;
}
```

```
void line1::setline1(double x1, double y1, double x2, double y2)
{
    this->setPoint1(x1, y1);
    this->setPoint2(x2, y2);
}
point1 line1::getPoint1()
{
    return this->point11;
}
point1 line1::getPoint2()
{
    return this->point2;
}

bool line1::equalTo()
{
    if (this->point11.getpointX() == this->point2.getpointX() &&
this->point11.getpointY() == this->point2.getpointY())
    {
        return true;
    }
    else
    {
        return false;
    }
}
std::ostream &operator<<(std::ostream &s, line1 &l)
{
    return s << "((" << l.getPoint1().getpointX() << "," <<
l.getPoint1().getpointY() << ")(" << l.getPoint2().getpointX() <<
"," << l.getPoint2().getpointY() << "))";
}

istream &operator>>(istream &in, line1 &l)
{
    cout << "Enter X and Y co-ordinates for Point 1 and Point 2\n";
    double x, y;
```

```
    in >> x;
    in >> y;
    double x1, y1;
    in >> x1;
    in >> y1;
    l.setline1(x, y, x1, y1);
    return in;
}
```

Main.cpp

```
#include <iostream>
#include "line1.cpp"
#include "point1.cpp"

using namespace std;

int main()
{
    point1 p1(10.0, 3.0);
    point1 p2(11.0, 3.0);
    line1 l1(p1, p2);

    point1 p3(p1);
    point1 p4(p2);
    line1 l2(p3, p4);
    cout << "line 1: " << l1;
    cout << "\nline 2 : " << l2 << "\n";
    // cout << l1.getPoint1().getpointX() << "\n";
    cout << "Distance : " << l1.Distance() << "\n";
    cout << "Slope : " << l1.Slope() << "\n";
    cout << "if both points are Equal (1 for true, 0 for False) : "
<< l1.equalTo() << "\n";
    // cout << ""l1 << endl;
    cin >> l1;
    cout << "New Line : " << l1 << endl;
    return 0;
}
```