EUROPEAN UNIVERSITY OF LEFKE
Faculty of Engineering
Department of Computer Engineering

COMP218
OBJECT-ORIENTED PROGRAMMING

Lab Work No. 7

Prepared by Seward Richard Mupereri (20140175)

Submitted to Dr. Ferhun Yorgancıoğlu

Task (1)

```
#include <iostream>
#include <iomanip>

using namespace std;

class Point {
public:
   Point();
   Point( int, int );
   Point( const Point& );
   ~Point();

   void set_x( int );
   void set_y( int );

   int get_x();
   int get_y();

   void print();

private:
   int x, y;

};

Point::Point() : x( 0 ), y( 0 ) {}

Point::Point( int x, int y )
{
   set_x( x );
   set_y( y );
}

Point::Point( const Point& copy ) : x( copy.x ), y( copy.y ) {}

Point::~Point(){}

void Point::set_x( int x )
{
   this->x = x;
}

void Point::set_y( int y )
{
   this->y = y;
}

int Point::get_x()
{
   return x;
}

int Point::get_y()
{
   return y;
}

void Point::print()
```

```
{
   cout << " POINT: [ " << x << ", " << y << " ]";
}

class Line {
public:
   Line();
   Line( Point, Point );
   Line( const Line& );
   ~Line();

   void set_point1( Point );
   void set_point2( Point );

   double get_slope();

   void print();

private:
   Point p1, p2;

};

Line::Line(){}

Line::Line ( Point p1, Point p2 )
{
   set_point1( p1 );
   set_point2( p2 );
}

Line::Line( const Line& copy ) : p1(copy.p1), p2(copy.p2) {}

Line::~Line() {}

void Line::set_point1( Point p )
{
   p1 = p;
}

void Line::set_point2( Point p )
{
   p2 = p;
}

double Line::get_slope()
{
   return ( (double) ( p1.get_y() - p2.get_y() ) / ( p1.get_x() - p2.get_x() ) );
}

void Line::print()
{
   cout << "       A line passing through        " << endl;
   p1.print();
   cout << "   and ";
   p2.print();
   cout << endl;
   cout << "           SLOPE = " << get_slope();
}

int main()
```

```
{
   int x, y;

   cout << "---------------------------------------" << endl;
   cout << "  CARTESIAN COORDINATE SYSTEM PROGRAM  " << endl;
   cout << "---------------------------------------" << endl;

   cout << setw(22) << "POINT 1" << endl;
   cout << setw(22) << "Enter x:";
   cin >> x;

   cout << setw(22) << "Enter y:";
   cin >> y;

   Point p1( x, y );

   cout << "---------------------------------------" << endl;

   cout << setw(22) << "POINT 2" << endl;
   cout << setw(22) << "Enter x:";
   cin >> x;

   cout << setw(22) << "Enter y:";
   cin >> y;

   Point p2( x, y );

   Line a( p1, p2 );

   cout << "---------------------------------------" << endl;

   a.print();

   cout << endl;
   cout << "---------------------------------------" << endl;
   cout << "           END OF PROGRAM              " << endl;
   cout << "---------------------------------------" << endl;

   return 0;
}
```

```
---------------------------------------
  CARTESIAN COORDINATE SYSTEM PROGRAM
---------------------------------------
               POINT 1
              Enter x:4
              Enter y:6
---------------------------------------
               POINT 2
              Enter x:2
              Enter y:9
---------------------------------------
        A line passing through
 POINT: [ 4, 6 ]   and  POINT: [ 2, 9 ]
             SLOPE = -1.5
---------------------------------------
            END OF PROGRAM
---------------------------------------
```

Task (2)

Header file – line.h

```
#ifndef LABWORK7_LINE_H
#define LABWORK7_LINE_H

class Point {
public:
   Point();
   Point( int, int );
   Point( const Point& );
   ~Point();

   void set_x( int );
   void set_y( int );

   int get_x();
   int get_y();

   void print();

private:
   int x, y;

};

class Line {
public:
   Line();
   Line( Point, Point );
   Line( const Line& );
   ~Line();

   void set_point1( Point );
   void set_point2( Point );
```

```
    double get_slope();

    void print();

private:
    Point p1, p2;

};
#endif //LABWORK7_LINE_H
```

Implementation file – line.cpp

```
#include <iostream>
#include "line.h"
using namespace std;

Point::Point() : x( 0 ), y( 0 ) {}

Point::Point( int x, int y )
{
   set_x( x );
   set_y( y );
}

Point::Point( const Point& copy ) : x( copy.x ), y( copy.y ) {}

Point::~Point(){}

void Point::set_x( int x )
{
   this->x = x;
}

void Point::set_y( int y )
{
   this->y = y;
}

int Point::get_x()
{
   return x;
}

int Point::get_y()
{
   return y;
}

void Point::print()
{
   cout << " POINT: [ " << x << ", " << y << " ]";
}



Line::Line(){}

Line::Line ( Point p1, Point p2 )
{
   set_point1( p1 );
   set_point2( p2 );
}

Line::Line( const Line& copy ) : p1(copy.p1), p2(copy.p2) {}

Line::~Line() {}

void Line::set_point1( Point p )
{
   p1 = p;
}
```

```
void Line::set_point2( Point p )
{
   p2 = p;
}

double Line::get_slope()
{
   return ( (double) ( p1.get_y() - p2.get_y() ) / ( p1.get_x() - p2.get_x() ) );
}

void Line::print()
{
   cout << "       A line passing through        " << endl;
   p1.print();
   cout << "   and ";
   p2.print();
   cout << endl;
   cout << "           SLOPE = " << get_slope();
}
```

Driver file – main.cpp

```
#include <iostream>
#include <iomanip>
#include "line.h"

using namespace std;

int main()
{
   int x, y;

   cout << "-------------------------------------" << endl;
   cout << "  CARTESIAN COORDINATE SYSTEM PROGRAM  " << endl;
   cout << "-------------------------------------" << endl;

   cout << setw(22) << "POINT 1" << endl;
   cout << setw(22) << "Enter x:";
   cin >> x;

   cout << setw(22) << "Enter y:";
   cin >> y;

   Point p1( x, y );

   cout << "-------------------------------------" << endl;

   cout << setw(22) << "POINT 2" << endl;
   cout << setw(22) << "Enter x:";
   cin >> x;

   cout << setw(22) << "Enter y:";
   cin >> y;

   Point p2( x, y );

   Line a( p1, p2 );

   cout << "-------------------------------------" << endl;

   a.print();

   cout << endl;
   cout << "-------------------------------------" << endl;
   cout << "           END OF PROGRAM            " << endl;
   cout << "-------------------------------------" << endl;

   return 0;
}
```

```
----------------------------------------
  CARTESIAN COORDINATE SYSTEM PROGRAM
----------------------------------------
               POINT 1
              Enter x:5
              Enter y:4
----------------------------------------
               POINT 2
              Enter x:8
              Enter y:9
----------------------------------------
        A line passing through
 POINT: [ 5, 4 ]   and  POINT: [ 8, 9 ]
             SLOPE = 1.66667
----------------------------------------
            END OF PROGRAM
----------------------------------------
```