Lecture Four: Pointers to Objects

Let's start with a different organization of our program:

```cpp
#include <iostream>
#include "Point.h"
#include "Line.h"

using namespace std;

int main() {
    Point a(3, 0), b(0, 4), c(1, 0), d(0, 1);
    Line u(c, d);
    cout << a.distanceTo(b) << endl;
    cout << u.length() << endl;
}
```

We have split the program in five files: two header files and three source code files.

```cpp
#ifndef POINT_H
#define POINT_H

class Point {
public:
    Point(double, double);
    double getX() const;
    double getY() const;
    double distanceTo(Point);
private:
    double x, y;
};
#endif
```

Point.h
```cpp
#include "Point.h"
#include <cmath>

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

double Point::getX() const { return x; }

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

double Point::distanceTo(Point other) {
    double dx = x - other.getX(), dy = y - other.getY();
    return sqrt(pow(dx, 2) + pow(dy, 2));
}

Point.cpp

#include "Line.h"
#include "Point.h"

Line::Line(Point a, Point b) : a(a), b(b) { }

double Line::length() { return a.distanceTo(b); }

Line.cpp

Finally here's how you can write the main program:

```cpp
#include <iostream>
#include "Point.h"
#include "Line.h"

using namespace std;

int main() {
    Point* a = new Point(3, 0);
    Point* b = new Point(0, 4);
    Point* c = new Point(1.4142, 0);
    Point* d = new Point(0, 1);
    Line* u = new Line(*c, *d);
    cout << a->distanceTo(*b) << endl;
    cout << (*u).length() << endl;
}

Notice the pointer notation.
Define a class Triangle. Objects of that type are supposed to be able to calculate their area.

```cpp
#define TRIANGLE_H

class Triangle {
public:
    Triangle(Point, Point, Point);
    double area();
private:
    Line a, b, c;
};

#endif

#include "Point.h"
#include "Line.h"
#include "Triangle.h"
#include <cmath>

Triangle::Triangle(Point a, Point b, Point c) : a(a, b), b(a, c), c(b, c) { }

double Triangle::area() {
    double s = (a.length() + b.length() + c.length()) / 2;
    return sqrt(s * (s - a.length()) * (s - b.length()) * (s - c.length()));
}

#include <iostream>
#include "Point.h"
#include "Line.h"
#include "Triangle.h"

using namespace std;

int main() {
    Point* a = new Point(3, 0);
    Point* b = new Point(0, 4);
    Point* c = new Point(0, 1); // area should be 4.5
    Triangle* t = new Triangle(*a, *b, *c);
    cout << t->area() << endl;
}
```

These were the only changes to what we had before.