

Lecture 6 Notes

Classes

New data types can be created using classes. A class contains a name for the new data type, as well as declarations of variables and functions which will exist in each instance of that data type.

Here is a class that represents a coordinate on a cartesian graph:

```
class Coordinate {
public:
    double x;
    double y;
};
```

The distance between two coordinates (x_1, y_1) and (x_2, y_2) is:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The following example uses our new data type to calculate the distance between two coordinates:

```
int main()
{
    Coordinate a, b;
    double distance;

    cout << "Enter x_1 and y_1: "s;
    cin >> a.x >> a.y;
    cout << "Enter x_2 and y_2: "s;
    cin >> b.x >> b.y;
    distance = sqrt(pow(b.x - a.x, 2) + pow(b.y - a.y, 2));
    cout << "The distance is: "s << distance << endl;
}
```

Calculating the distance between two coordinates is a common operation, so we can make a function inside our class to do that for us:

```
class Coordinate {
public:
    double x;
    double y;

    double distance_from(const Coordinate& c) const
    {
        return sqrt(pow(x - c.x, 2) + pow(y - c.y, 2));
    }
};
```

The keyword “const” after the function declaration denotes that this function does not modify any of the variables inside the class.

Now our main function can be rewritten as follows:

```
int main()
{
    Coordinate a, b;
    cout << "Enter x_1 and y_1: "s;
    cin >> a.x >> a.y;
    cout << "Enter x_2 and y_2: "s;
    cin >> b.x >> b.y;
    cout << "The distance is: "s << a.distance_from(b) << endl;
}
```

Scoping

The label “public:” inside a class is followed by declarations of variables and functions which can be accessed from code that exists outside the class.

The label “private:” means the following declarations are only accessible from code existing inside the class.

Listing 1: Example of a class that has private members

```
class Line {
private:
    Coordinate a, b;
public:
    void set_coordinates(const Coordinate& _a, const Coordinate& _b)
    {
        a = _a;
        b = _b;
    }
    double length() const
    {
        return a.distance_from(b);
    }
    double slope() const
    {
        return (b.y - a.y) / (b.x - a.x);
    }
};
```

This data type “Line” has two private member variables a and b. The public member functions length and slope do not modify a and b (hence their “const” suffix). The public member function set_coordinates directly assigns to a and b without exposing their current values.

– *Mark Swoope*