apart. That uncertainty is no more desirable in a math function than it is in a wilderness
guide.)

Given these formulas, you can write the function as follows:

```cpp
// convert rectangular to polar coordinates
polar rect_to_polar(rect xypos) // type polar
{
    polar answer;

    answer.distance =
        sqrt( xypos.x * xypos.x + xypos.y * xypos.y);
    answer.angle = atan2(xypos.y, xypos.x);
    return answer; // returns a polar structure
}
```

Now that the functions are ready, writing the rest of the program is straightforward. Listing 7.12 presents the result.

**Listing 7.12 strctfun.cpp**

```cpp
// strctfun.cpp -- functions with a structure argument
#include <iostream>
#include <cmath>
using namespace std;

// structure templates
struct polar
{
    double distance; // distance from origin
    double angle;   // direction from origin
};

struct rect
{
    double x;       // horizontal distance from origin
    double y;       // vertical distance from origin
};
```