    return x * x * x; // a type double value
}

For example, the function call `cube(1.2)` returns the value 1.728. Note that this return statement uses an expression. The function computes the value of the expression (1.728, in this case) and returns the value.

### Function Prototyping and Function Calls

By now you are familiar with making function calls, but you may be less comfortable with function prototyping because that's often been hidden in the `include` files. Let's use the `cheers()` and `cube()` functions in a program (see Listing 7.2); notice the function prototypes.

**Listing 7.2 protos.cpp**

    // protos.cpp -- use prototypes and function calls
    #include <iostream>
    using namespace std;
    void cheers(int);       // prototype: no return value
    double cube(double x);  // prototype: returns a double
    int main(void)
    {
        cheers(5);          // function call
        cout << "Give me a number: ";
        double side;
        cin >> side;
        double volume = cube(side); // function call
        cout << "A " << side << "-foot cube has a volume of ";
        cout << volume << " cubic feet.\n";
        cheers(cube(2));    // prototype protection at work
        return 0;
    }

    void cheers(int n)