Here's the output:

main() will call the simple() function:
I'm but a simple function.

Let's take a more detailed look at these steps now.

**Defining a Function**

You can group functions into two categories: those that don't have return values and those that do. Functions without return values are termed type **void** functions and have the following general form:

```cpp
void functionName(parameterList)
{
    statement(s)
    return; // optional
}
```

Here `parameterList` specifies the types and number of arguments (parameters) passed to the function. This chapter more fully investigates this list later. The optional return statement marks the end of the function. Otherwise, the function terminates at the closing brace. Type **void** functions correspond to Pascal procedures, FORTRAN subroutines, and modern BASIC subprogram procedures. Typically, you use a **void** function to perform some sort of action. For example, a function to print Cheers! a given number \( n \) of times can look like this:

```cpp
void cheers(int n) // no return value
{
    for (int i = 0; i < n; i++)
        cout << "Cheers! ";
    cout << "\n";
}
```

The `int n` parameter list means that `cheers()` expects to be passed an `int` value as an argument when you call this function.