*Stubs* and *drivers* are very helpful tools for testing and debugging programs that use functions. They allow you to test the individual functions in a program, in isolation from the parts of the program that call the functions.

A *stub* is a dummy function that is called instead of the actual function it represents. It usually displays a test message acknowledging that it was called, and nothing more. For example, if a stub were used for the `showFees` function in Program 6-10 (the modular health club membership program), it might look like this:

```
void showFees(double membershipRate, int months)
{
     cout << "The showFees fonction was called with "
          << "the following arguments:\n"
          << "memberRate: " << memberRate << endl
          << "months: " << months << endl;
}
```

The following is an example output of the program if it were run with the stub instead of the actual `showFee` function. (A version of the health club program using this stub function is on the Student CD. The program is named `HealthClubWithStub.cpp`).

```
                    Health Club Membership Menu

1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program

Enter your choice: 1 [Enter]
For how many months? 4 [Enter]
The showFees function was called with the following arguments:
memberRate: 40.00
months: 4

                    Health Club Membership Menu

1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program

Enter your choice: 4 [Enter]
```

As you can see, by replacing an actual function with a stub, you can concentrate your testing efforts on the parts of the program that call the function. Primarily, the stub allows you to determine whether your program is calling a function when you expect it to, and confirm that valid values are being passed to the function. If the stub represents a function that returns a value, then the stub should return a test value. This helps you confirm that a return value is being handled properly.When the parts of the program that call a function are debugged to your satisfaction, you can move on to testing and debugging the actual functions themselves. This is where *drivers* become useful.

A driver is a program that tests a function by simply calling it. If the function accepts arguments, the driver passes test data. If the function returns a value, the driver displays the return value on the screen. This allows you to see how the function performs in isolation from the rest of the program it will eventually be a part of. Program 6-29 shows a driver for testing the `showFees` function in the health club membership program.

**Program 6-29**

```
// This program is a driver for testing the showFees function.
#include <iostream>
using namespace std;

// Prototype
void showFees(double, int);

int main()
{
      // Constants for membership rates
      const double ADULT = 40.0;
      const double SENIOR = 30.0;
      const double CHILD = 20.0;

      // Perform a test for adult membership.
      cout << “Testing an adult membership…\n”
           << “Calling the showFees function with arguments “
           << ADULT << “ and 10.\n”;
      showFees(ADULT, 10);

      // Perform a test for senior citizen membership.
      cout << “\nTesting a senior citizen membership…\n”
           << “Calling the showFees function with arguments “
           << SENIOR << “ and 10.\n”;
      showFees(SENIOR, 10);

      // Perform a test for child membership.
      cout << “\nTesting a child membership…\n”
           << “\nCalling the showFees function with arguments “
           << CHILD << “ and 10.\n”;
      showFees(CHILD, 10);
      return 0;
}
```

**Program Output**

```
Testing an adult membership…
Calling the showFees function with arguments 40 and 10.
The total charges are $400

Testing a senior citizen membership…
Calling the showFees function with arguments 30 and 10.
The total charges are $300

Testing a child membership…

Calling the showFees function with arguments 20 and 10.
```

```
The total charges are $200
```

As shown in Program 6-29, a driver can be used to thoroughly test a function. It can repeatedly call the function with different test values as arguments. When the function performs as desired, it can be placed into the actual program it will be part of.

*This rubbish copied from* Starting out with C++: From Control Structures through Objects, Fifth Edition *by Tony Gaddis, page 359-362. As much original formatting is preserved as is practical.*