Overview

Up to now you have written programs with only a main function. In Chapter 3 you learned that programs are often divided into more than one function. Programs are divided up so that the functions perform a specific task or *function*. You will learn how to create functions in this chapter as well as how to use the pre-written functions that are included with your compiler.

CHAPTER 9, SECTION 1

How to Build Programs with Functions

Examine the source code in Figure 9-1. The program consists of one function, **main()**. You may have difficulty, however, quickly determining what the program accomplishes.

When the program is run, the user is prompted from a menu to choose to view a series of numbers. Depending upon the user's choice, the program displays a series of odd numbers, even numbers, or all integers from 1 to 30.

Let's run the program to see its output.

EXERCISE 9-1 SINGLE-FUNCTION PROGRAM

1. Retrieve the source file *SERIES.CPP*.
2. Compile and run the program to see the program's output.
3. Close the source code file.

The program you just executed could have been better built using more than one function. The diagram in Figure 9-2, known as a ***Visual Table of Contents (VTOC)***, illustrates the point. The lines represent connections between functions. Each function can be accessed by the function above it as long as they are connected by a line.

In this case, the main function "calls" the **get_choice** function to ask the user to choose the series to display. Next, the **handle_choice** function is called to direct the program flow to one of the three functions under it—one for each series. The source code for Figure 9-2 is presented later in this chapter.

GUIDELINES FOR BUILDING A PROGRAM WITH FUNCTIONS

You may recall from Chapter 2 that good programmers follow the five basic steps shown below when developing programs.

1. Define the problem.
2. Develop an algorithm.
3. Code the program.

```
#include<iostream.h>

main()
{
 int choice;  // variable for user input
 int i;       // variable for loops and output

 do  // loop until a valid choice is entered
  {
   cout << "Which series do you wish to display?\n";
   cout << "1 - Odd numbers from 1 to 30\n";
   cout << "2 - Even numbers from 1 to 30\n";
   cout << "3 - All numbers from 1 to 30\n";
   cin >> choice;  // get choice from user
   if ((choice < 1) || (choice > 3))  // if invalid entry, give message
    {
     cout << "Choice must be 1, 2, or 3\n";
    }
  } while ((choice < 1) || (choice > 3));

  switch (choice)
    {
     case 1:
       for (i = 1; i <= 30; i = i + 2)
       cout << i << ' ';
       cout << endl;
       break;
     case 2:
       for (i = 2; i <= 30; i = i + 2)
       cout << i << ' ';
       cout << endl;
       break;
     case 3:
       for (i = 1; i <= 30; i++)
       cout << i << ' ';
       cout << endl;
       break;
    }
  return 0;
}
```

FIGURE 9-1
This entire program is in the main function.

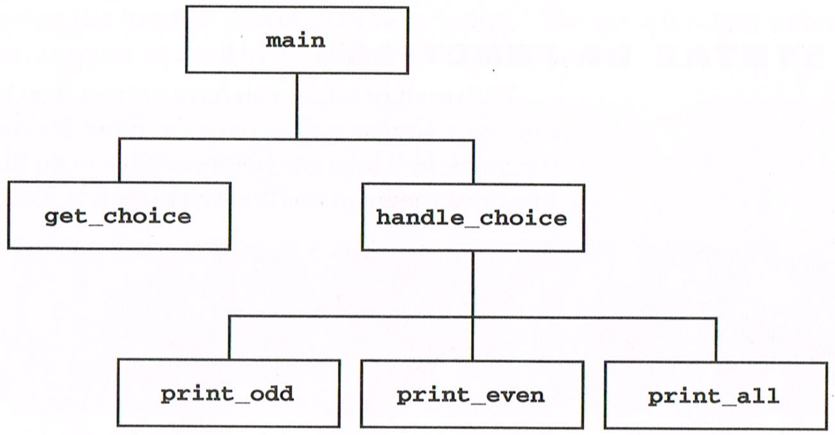

FIGURE 9-2
A diagram that shows the functions of a program is sometimes called a Visual Table of Contents.