

You are probably now beginning to see why C++ is so powerful—and so dangerous. A statement like the one above should be avoided because it fails to indicate the specific variable that is being changed. Although C++ programmers are given the freedom to work with memory in a rich variety of ways, pointers should be used only when they improve the program.

Consider the program in Figure 10-3. The program declares and initializes two floating-point numbers. Using a do while loop, the program repeatedly picks the larger of the two numbers and divides it by 2. The loop ends when one of the variables becomes less than 1.

The program uses a pointer to provide an efficient solution. By setting a pointer to point to the larger of the two values, the larger value can be printed and divided by 2 by use of the pointer, rather than the variable itself. By using a pointer, the same code can be used no matter which variable is the larger.

```
#include <iostream.h> // necessary for cout command

int main()
{
    float a, b;           // declare two floating-point numbers
    float *float_ptr;     // declare a pointer to a float

    a = 169.8;
    b = 237.5;

    do
    {
        cout << "The two numbers are " << a << " and " << b << endl;

        if (a >= b)
        {
            float_ptr = &a;
        }
        else
        {
            float_ptr = &b;
        }

        cout << "The largest of the two numbers is "
            << *float_ptr << endl;
        cout << *float_ptr;
        *float_ptr = *float_ptr / 2;
        cout << " divided by 2 is " << *float_ptr << endl;
    } while((a > 1.0) && (b > 1.0));
    return 0;
}
```

FIGURE 10-3
This program accesses variables by use of a pointer.

EXERCISE 10-3 USING POINTERS

1. Enter the program shown in Figure 10-3. Save the source code file as *THEPOINT.CPP*.
2. Study the source code closely before you run the program. Compile and run the program to see its output. The program divides the larger of the 2 values by 2 until one of the values becomes less than 1.
3. Close the source code file.

On the Net

Learn more about pointers and see more examples at <http://www.ProgramCPP.com>. See topic 10.1.1.

SECTION 10.1 QUESTIONS

1. What is stored in a pointer?
2. What symbol is used for the address-of operator?
3. Write a statement that declares a pointer to a variable of type double.
4. Write a statement that assigns the pointer you declared in question 3 to the address of a variable named x.
5. Now change the value of the variable x to 9.9 using the pointer you declared and the dereferencing operator.

PROBLEM 10.1.1

Write a program that declares two variables of type float and a single pointer of type float. First initialize the pointer to point to the first float variable. Use the dereferencing operator to initialize the variable to 1.25. Next point the pointer to the second float variable and use the pointer to initialize the variable to 2.5. Print the value of the first variable to the screen by accessing the variable directly. Print the second variable using the dereferencing operator. Save the source code file as *POINTER2.CPP*.

CHAPTER 10, SECTION 2

More About Character Arrays

Now that you have been exposed to the basics of pointers, some matters relating to character arrays may make more sense. In this section, you will strengthen your knowledge about character arrays.