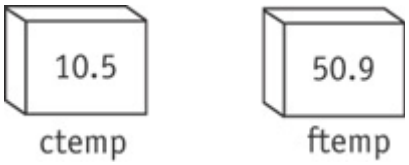
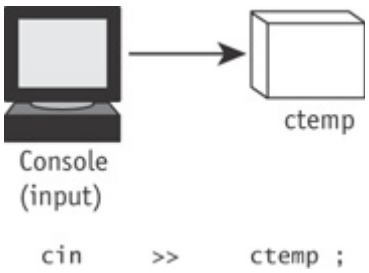
Storing Data: C++ Variables

If all you could do was print messages, C++ wouldn't be useful. The point is usually to get new data from somewhere—such as end-user input—and then do something interesting with it.

Such operations require *variables*; these are locations into which you can place data. You can think of variables as magic boxes that hold values. As the program proceeds, it can read, write, or alter these values as needed. The upcoming example uses variables named ctemp and ftemp to hold Celsius and Fahrenheit values, respectively.



How do values get put into variables? One way is through console input. In C++, you can input values by using the **cin** object, representing (appropriately enough) console input. With **cin**, you use a stream operator showing data flowing to the right (>>):



Here's what happens in response to this statement. (The actual process is a little more complicated, but don't worry about that for now.)

1. The program suspends running and waits for the user to enter a number.
2. The user types a number and presses ENTER.
3. The number is accepted and placed in the variable ctemp (in this case).
4. The program resumes running.

So, if you think about it, a lot happens in response to this statement:

```
cin >> ctemp;
```

But before you can use a variable in C++, you must declare it. This is an absolute rule, and it makes C++ different from Basic, which is sloppy in this regard and doesn't require declaration. (But generations of Basic programmers have banged their heads against their terminals as they discovered errors cropping up as a result of Basic's laxness about variables.)

This is important enough to justify restating, so I'll make it a cardinal rule:

In C++, you must declare a variable before using it.

To declare a variable, you have to first know what *data type* to use. This is a critical concept in C++ as in most other languages.