The beginning of Chapter 3, "Dealing with Data," mentions three fundamental properties of which a computer program must keep track when it stores data. To save the book the wear and tear of your thumbing back to that chapter, here are those properties again:

- Where the information is stored
- What value is kept there
- What kind of information is stored

You've used one strategy for accomplishing these ends: defining a simple variable. The declaration statement provides the type and a symbolic name for the value. It also causes the program to allocate memory for the value and to keep track of the location internally.

Let's look at a second strategy now, one that becomes particularly important in developing C++ classes. This strategy is based on pointers, which are variables that store addresses of values rather than the values themselves. But before discussing pointers, let's see how to find addresses explicitly for ordinary variables. Just apply the address operator, represented by &, to a variable to get its location; for example, if home is a variable, &home is its address. Listing 4.9 demonstrates this operator.

**Listing 4.9 address.cpp**

```cpp
// address.cpp _ using the & operator to find addresses
#include <iostream>
using namespace std;
int main()
{
    int donuts = 6;
    double cups = 4.5;

    cout << "donuts value = " << donuts;
    cout << " and donuts address = " << &donuts << "\n";
    // NOTE: you may need to use unsigned (&donuts)
    //    and unsigned (&cups)
    cout << "cups value = " << cups;
    cout << " and cups address = " << &cups << "\n";
```