Variables, including parameters, declared within a function are private to the function. When a function is called, the computer allocates the memory needed for these variables. When the function terminates, the computer frees the memory that was used for those variables. (Some C++ literature refers to this allocating and freeing of memory as creating and destroying variables. That does make it sound much more exciting.) Such variables are called **local variables** because they are localized to the function. As we mentioned, this helps preserve data integrity. It also means that if you declare a variable called `x` in `main()` and another variable called `x` in some other function, these are two distinct, unrelated variables, much as the Albany in California is distinct from the Albany in New York. (See Figure 7.3.) Such variables also are termed automatic variables because they are allocated and deallocated automatically during program execution.

**Figure 7.3. Local variables.**