CHAPTER 13

STRINGS

In this WantMore file, you will learn a string manipulation technique that was not covered in Chapter 13. More specifically, you will learn how to use the `transform` function to convert a string to a different case.

THE `transform` FUNCTION

You can use the C++ `transform` function to convert the contents of a `string` variable to either uppercase or lowercase. Figure WM-Figure 13-1 shows the function's syntax and includes two examples of using the function in a C++ statement. The `transform` function is defined in the algorithm file, so you must include the `#include <algorithm>` directive in any program that uses the function.

transform Function

Syntax

transform(*string***.begin(),** *string***.end(),** *string***.begin(),** *function***);**

Example 1

```
string city = "London";
transform(city.begin(), city.end(), city.begin(), toupper);
```

transforms (converts) the contents of the `city` variable to uppercase; after the function is processed, the `city` variable will contain the string "LONDON"

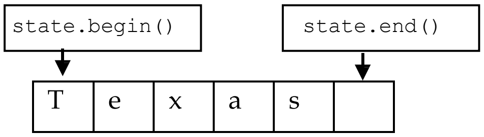
Example 2

```
string state = "Texas";
transform(state.begin(),state.end(),state.begin(), tolower);
```

transforms (converts) the contents of the `state` variable to lowercase; after the function is processed, the `state` variable will contain the string "texas"

WM-Figure 13-1 Syntax and examples of the `transform` function

In the syntax, *string* is the name of a `string` variable that contains the string you want converted, or transformed, to either uppercase or lowercase. The first two arguments in the function specify the range of characters to transform in the *string*. To transform the entire contents of a `string` variable, you use *string*`.begin()` as the first argument and use *string*`.end()` as the second argument. The *string*`.begin()` refers to the first character in the `string` variable, and *string*`.end()` refers to the location that is just past the end of the `string` variable in the computer's internal memory. For example, if a `string` variable named `state` contains the string "Texas", then `state.begin ()` refers to the letter T, and `state.end()` refers to the memory location following the letter s, as illustrated in WM-Figure 13-2.



WM-Figure 13-2 Illustration of `state.begin()` and `state.end()`

The `transform` function converts (transforms) each of the characters contained in the range specified in the function's first two arguments. The range begins with the character whose location is specified in the first argument. From there, the range continues up to, but not including, the character whose location is specified in the second argument. The function stores the results of the conversion beginning in the location specified in its third argument, replacing the characters currently stored at that location. For example, the third argument shown in the function's syntax, *string*`.begin()`, tells the function to store the transformed string in the `string` variable, beginning with the first character in the variable.

The last argument in the `transform` function's syntax is the name of a function. The function indicates the task to be performed on the string contained in the `string` variable. To transform the string to uppercase, you use `toupper` as the function argument. To transform the string to lowercase, you use `tolower` as the function argument. Be sure to use `toupper` and `tolower`, rather than `toupper()` and `tolower()`. Including the parentheses after `toupper` or `tolower` in the transform function will produce a syntax error when the program is compiled.