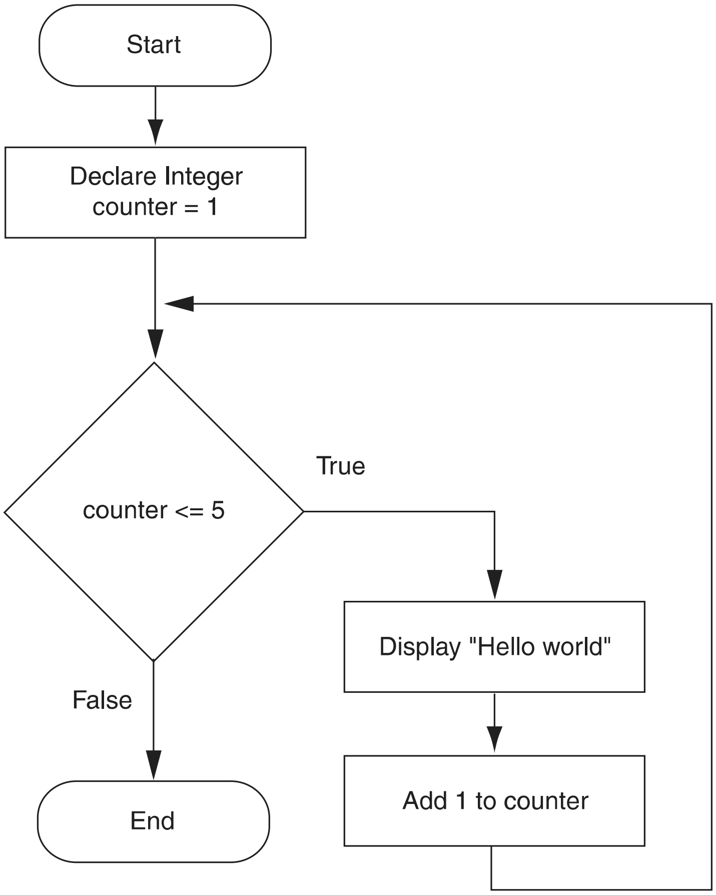
In the flowchart, assume that counter is an Integer variable. The first step is to set counter to the appropriate starting value. Then, determine whether counter is less than or equal to the maximum value. If this is true, the body of the loop executes. Otherwise, the program exits the loop. Notice that in the body of the loop one or more statements are executed, and then 1 is added to counter.

For example, look at the flowchart in Figure 5-13. First, an Integer variable named counter is declared and initialized with the starting value 1. Then, the expression counter <= 5 is tested. If this expression is true the message "Hello world" is displayed and 1 is added to counter. Otherwise, the program exits the loop. If you follow the logic of this program you will see that the loop will iterate five times.

**Figure 5-13** A count-controlled loop



## The For Statement

VideoNote
The For Statement

Count-controlled loops are so common in programming that most languages provide a statement just for them. This is usually called the For statement. The For statement is specifically designed to initialize, test, and increment a counter variable. Here is the general format that we will use to write the For statement in pseudocode:

```
For counterVariable = startingValue To maxValue
     statement
     statement
     statement
     etc.
End For
```

These statements are the body of the loop.

In the general format, *counterVariable* is the name of a variable that is used as a counter, *startingValue* is the value that the counter will be initially set to, and *maxValue* is the maximum value that the counter can contain. When the loop executes, the following actions are performed:

1. The *counterVariable* is set to the *startingValue*.
2. The *counterVariable* is compared to the *maxValue*. If the *counterVariable* is greater than *maxValue*, the loop stops. Otherwise:
   a. The statements that appear in the body of the loop are executed.
   b. The *counterVariable* is incremented.
   c. The loop starts over again at Step 2.

An actual `For` loop is easy to understand, so let's look at one. The pseudocode in Program 5-8 uses a `For` loop to display "Hello world" five times. The flowchart in Figure 5-14 shows the logic of the program.

**Program 5-8**

```
1 Declare Integer counter
2 Constant Integer MAX_VALUE = 5
3
4 For counter = 1 To MAX_VALUE
5    Display "Hello world"
6 End For
```

**Program Output**

```
Hello world
Hello world
Hello world
Hello world
Hello world
```

Line 1 declares an `Integer` variable that will be used as the counter variable. You do not have to name the variable `counter` (you are free to name it anything you wish), but in many cases that is an appropriate name. Line 2 declares a constant named `MAX_VALUE` that will be used as the counter's maximum value. The `For` loop begins in line 4 with the statement `For counter = 1 To MAX_VALUE`. This specifies that the `counter` variable will start with the value 1 and will end with the value 5. At the end of each loop iteration, the `counter` variable will be incremented by 1, so this loop will iterate five times. Each time it iterates, it displays "Hello world."

Notice that the loop does not contain a statement to increment the `counter` variable. This happens automatically in a `For` loop, at the end of each iteration. For that