Both the for loop and the while loop have bodies consisting of the single statement following the parenthesized expressions. As you've seen, that single statement can be a block, which can contain several statements. Keep in mind that braces, not indentation, define a block. Consider the following loop, for example:

```cpp
i = 0;
while (name[i] != '\0')
    cout << name[i] << "\n";
i++;
cout << "Done\n";
```

The indentation tells us the program author intended the i++; statement to be part of the loop body. The absence of braces, however, tells the compiler that the body consists solely of the first cout statement. Thus, the loop keeps printing the first character of the array indefinitely. The program never reaches the i++; statement because it is outside the loop.

The next example shows another potential pitfall:

```cpp
i = 0;
while (name[i] != '\0')  // problem semicolon
{
    cout << name[i] << "\n";
i++;
}
cout << "Done\n";
```

This time the code got the braces right, but it also inserted an extra semicolon. Remember, a semicolon terminates a statement, so this semicolon terminates the while loop. In other words, the body of the loop is a null statement, that is, nothing followed by a semicolon. All the material in braces now comes after the loop. It never is reached. Instead, the loop cycles doing nothing forever. Beware the