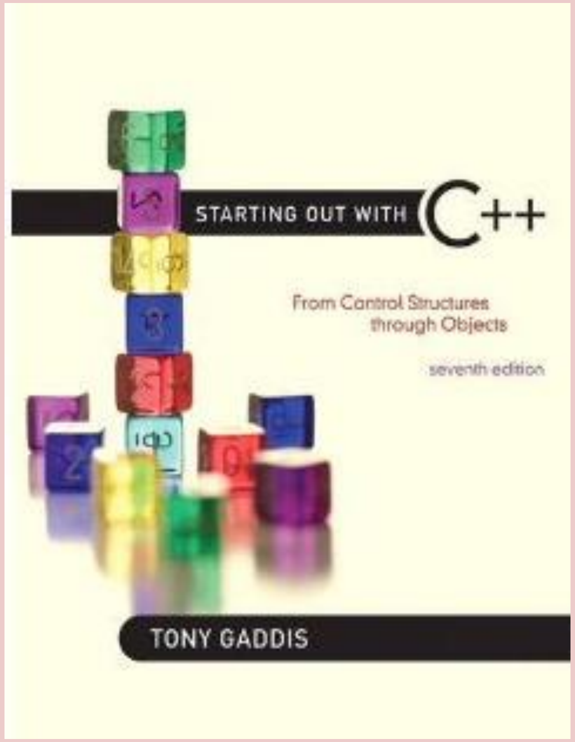
Software Design & Programming I



Starting Out with C++ (From Control Structures through Objects) 7th Edition
Written by: Tony Gaddis
Pearson - Addison Wesley
ISBN: 13-978-0-132-57625-3

Chapter 5

Looping

The Increment and Decrement Operators

C++ provides a pair of unary operators for incrementing and decrementing variables.

To increment a value means to increase it, and to decrement a value means to decrease it. In the example below, qtyOrdered is incremented by 10 and numSold is decremented by 3.

```
qtyOrdered = qtyOrdered + 10;
numSold = numSold  3;
```

Although the values stored in variables can be increased or decreased by any amount, it is particularly common to increment them or decrement them by 1. In fact, if we say that a value is being incremented or decremented without specifying by how much, it is understood that it is being incremented or decremented by 1.

The Increment and Decrement Operators

C++ provides a pair of unary operators to do this. The ++ operator increases its operand's value by 1. The – – operator decreases its operand's value by 1.

Here are three different ways to increment the value of the variable num by 1.

```
num = num + 1;
num += 1;
num++;
```

And here are three different ways to decrement it by 1:

```
num = num - 1;
num -= 1;
num--;
```

NOTE: The expression num++ is pronounced "num plus plus," and num- - is pronounced "num minus minus."

The Increment and Decrement Operators

Our examples so far show the increment and decrement operators used in postfix mode, which means the operator is placed after the variable. The operators also work in prefix mode, where the operator is placed before the variable name:

```
++num;
--num;
```

In both prefix and postfix mode, these operators add 1 to, or subtract 1 from, their operand.

The Increment and Decrement Operators

The following example illustrates the use of these operators in both prefix and postfix mode. Notice that there is no space between the name of the variable and the ++ or – – preceding it or following it.

```
num = 4;
num++;                  // now num has the value 5
++num;                  // now num has the value 6
num--;                  // now num has the value 5 again
--num;                  // now num has the value 4 again
```

The Difference Between Postfix and Prefix Modes

In the simple statements used in Program 5-1, it doesn’t matter if the increment or decrement operator is used in postfix or prefix mode. The difference is important, however, when these operators are used in statements that do more than just increment or decrement.
For example, look at the following lines:

```
num = 4;
cout << num++;
```

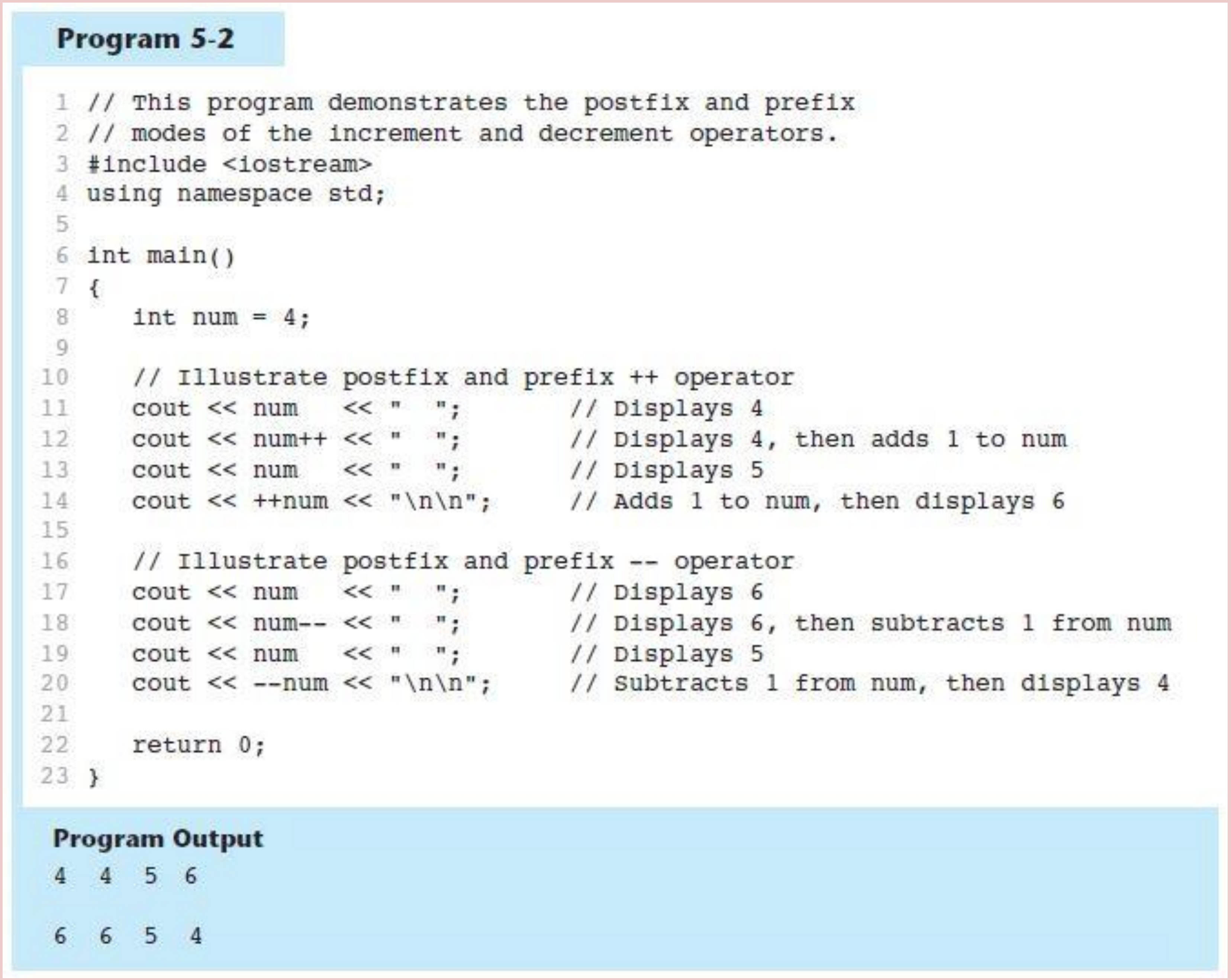
This cout statement is doing two things: 1) displaying the value of num, and 2) incrementing num. But which happens first? cout will display a different value if num is incremented first than if num is incremented last. The answer depends on the mode of the increment operator.

The Difference Between Postfix and Prefix Modes

Postfix mode causes the increment to happen after the value of the variable is used in the expression. In the example, cout will display 4, then num will be incremented to 5. Prefix mode, however, causes the increment to happen first. In the following statements, num will first be incremented to 5, then cout will display 5:

```
num = 4;
cout << ++num;
```

The Difference Between Postfix and Prefix Modes



Program 5-2

```
// This program demonstrates the postfix and prefix
// modes of the increment and decrement operators.
#include <iostream>
using namespace std;

int main()
{
   int num = 4;

   // Illustrate postfix and prefix ++ operator
   cout << num   << "  ";         // Displays 4
   cout << num++ << "  ";         // Displays 4, then adds 1 to num
   cout << num   << "  ";         // Displays 5
   cout << ++num << "\n\n";       // Adds 1 to num, then displays 6

   // Illustrate postfix and prefix -- operator
   cout << num   << "  ";         // Displays 6
   cout << num-- << "  ";         // Displays 6, then subtracts 1 from num
   cout << num   << "  ";         // Displays 5
   cout << --num << "\n\n";       // Subtracts 1 from num, then displays 4

   return 0;
}
```

Program Output

```
4  4  5  6

6  6  5  4
```

The Difference Between Postfix and Prefix Modes

Example, look at the following code:

```
int x = 1;
int y
y = x++;                    // Postfix increment
```

The first statement defines the variable x (initialized with the value 1) and the second statement defines the variable y. The third statement does two things:

- It assigns the value of x to the variable y.
- The variable x is incremented.

The Difference Between Postfix and Prefix Modes

Let's look at the same code, but with the ++ operator used in prefix mode:

```
int x = 1;
int y;
y = ++x;                    // Prefix increment
```

In the third statement, the ++ operator is used in prefix mode, causing variable x to be incremented before the assignment takes place. So, this code will store 2 in y. After the code has executed, x and y will both contain 2.

Using ++ and -- in Mathematical Expressions

The increment and decrement operators can also be used on variables in mathematical expressions. Consider the following program segment:

```
a = 2;
b = 5;
c = a * b++;
cout << a << " " << b << " " << c;
```

In the statement c = a * b++, c is assigned the value of a times b, which is 10. The variable b is then incremented. The cout statement will display

```
2     6     10
```

Using ++ and -- in Mathematical Expressions

If the statement were changed to read

```
c = a * ++b;
```

the variable b would be incremented before it was multiplied by a. In this case c would be assigned the value of 2 times 6, so the cout statement would display

```
2      6      12
```

You can pack a lot of action into a single statement using the increment and decrement operators, but don’t get too tricky with them. You might be tempted to try something like the following, thinking that c will be assigned 11:

```
a = 2;
b = 5;
c = ++(a * b); // Error!
```

Using ++ and -- in Relational Expressions

The ++ and -- operators may also be used in relational expressions. Just as in mathematical expressions, the difference between postfix and prefix mode is critical. Consider the following program segment:

```
x = 10;
if (x++ > 10)
cout << "x is greater than 10.\n";
```

Two operations are happening in this if statement: 1) The value in x is tested to determine if it is greater than 10, and 2) x is incremented. Because the increment operator is used in postfix mode, the comparison happens first. Since 10 is not greater than 10, the cout statement won’t execute.

Using ++ and -- in Relational Expressions

If the mode of the increment operator is changed, however, the if statement will compare 11 to 10 and the cout statement will execute:

```
x = 10;
if (++x > 10)
cout << "x is greater than 10.\n";
```

Introduction to Loops: The while Loop

-

The do-while Loop

-

The for Loop

-