

INCREMENT AND DECREMENT OPERATORS*

Kenneth Leroy Busbee

This work is produced by OpenStax-CNIX and licensed under the Creative Commons Attribution License 3.0[†]

Abstract

An introduction to the increment and decrement operators used in programming with details of C++ usage.

1 General Discussion

The idea of increment or decrement is to either add or subtract 1 from a variable that is usually acting as a flag. Using a variable named counter; in generic terms, for example:

increment the counter

The concept is:

counter is assigned counter + 1

That is you fetch the existing value of the counter and add one then store the answer back into the variable counter. Many programming languages allow their increment and decrement operators to only be used with the integer data type. Programmers will sometimes use inc and dec as abbreviations for increment and decrement respectively.

Operator symbols and/or names vary with different programming languages. The C++ programming language operators with their meanings are:

C++ Operator	Meaning
++	increment, two plus signs
-	decrement, two minus signs

Table 1

2 C++ Code Examples

2.1 Basic Concept

Within the C++ programming language the increment and decrement are often used in this simple generic way. The operator of increment is represented by two plus signs in a row. Examples:

*Version 1.4: Jan 15, 2010 1:22 pm -0600

[†]<http://creativecommons.org/licenses/by/3.0/>

```
counter = counter + 1;
counter += 1;
counter++;
++counter;
```

As C++ statements, the four examples all do the same thing. They add 1 to the value of whatever is stored in counter. The decrement operator is represented by two minus signs in a row. They would subtract 1 from the value of whatever was in the variable being decremented. The precedence of increment and decrement depends on if the operator is attached to the right of the operand (postfix) or to the left of the operand (prefix). Within C++ postfix and prefix do not have the same precedence.

2.2 Postfix Increment

Postfix increment says to use my existing value then when you are done with the other operators; increment me. An example:

```
int oldest = 44; // variable set up with initialization
    then later on in the code
age = oldest++;
```

The first use of the oldest variable is an Rvalue context where the existing value of 44 is pulled or fetched and then assigned to the variable age; then the variable oldest is incremented with its value changing from 44 to 45. This seems to be a violation of precedence because increment is higher precedence than assignment. But that is how postfix increment works within the C++ programming language.

2.3 Prefix Increment

Prefix increment says to increment me now and use my new value in any calculation. An example:

```
int oldest = 44; // variable set up with initialization
    then later on in the code
age = ++oldest;
```

The variable oldest is incremented with the new value changing it from 44 to 45; then the new value is assigned to age.

In postfix age is assigned 44 in prefix age is assigned 45. One way to help remember the difference is to think of postfix as being polite (use my existing value and return to increment me after the other operators are done) where as prefix has an ego (I am important so increment me first and use my new value for the rest of the evaluations).

2.4 Allowable Data Types

Within some programming languages, increment and decrement can be used only on the integer data type. C++ however, expands this not only to all of the integer family but also to the floating-point family (float and double). Incrementing 3.87 will change the value to 4.87. Decrementing 'C' will change the value to 'B'. Remember the ASCII character values are really one byte unsigned integers (domain from 0 to 255).

2.5 Exercises

Exercise 1

(Solution on p. 4.)

Evaluate the following items using increment or decrement:

1. True or false: $x = x + 1$ and $x += 1$ and $x++$ all accomplish increment?
2. Given: `int y = 19;` and `int z;` what values will y and z have after: `z = y--;`
3. Given: `double x = 7.77;` and `int y;` what values will x and y have after: `y = ++x;`
4. Is this ok? Why or why not? `6 * ++(age -3)`

3 Definitions

Definition 1: increment

Adding one to the value of a variable.

Definition 2: decrement

Subtracting one from the value of a variable.

Definition 3: postfix

Placing the increment or decrement operator to the right of the operand.

Definition 4: prefix

Placing the increment or decrement operator to the left of the operand.

Solutions to Exercises in this Module

Solution to Exercise (p. 3)

Answers:

1. true
2. y is: 18 and z is: 19
3. x is: 8.77 and y is: 8 Note: truncation of 8.77 to 8 upon demotion.
4. Not ok. Error, the item incremented must have Lvalue attributes, usually a variable. It is an expression not a variable.