**Multiple and Combined Assignments – Gaddis reference pg.101-103**

**CONCEPT: Multiple assignment means to assign the same value to several variables with one statement.**

C++ allows you to assign a value to multiple variables at once. If a program has several variables, such as a, b, c, and d, and each variable needs to be assigned a value, such as 12, the following statement may be constructed:

a = b = c = d = 12;

The value 12 will be assigned to each variable listed in the statement. This works because the assignment operations are carried out from right to left. First 12 is assigned to d. Then d’s value, now a 12, is assigned to c. Then c’s value is assigned to b, and finally b’s value is assigned to a.

Quite often programs have assignment statements of the following form: number = number + 1;  
The expression on the right side of the assignment operator gives the value of number plus 1. The result is then assigned to number, replacing the value that was previously stored there. Effectively, this statement adds 1 to number.

Table 3-8 shows other examples of statements written this way.

**Statement What It Does Value of x after the Statement**

x=x+4; Adds 4 to x 10

x = x - 3; Subtracts 3 from x 3

x= x \* 10; Multiplies x by 10 60

x = x / 2; Divides x by 2 3

x = x % 4 Makes x the remainder of x/4 2

Because these types of operations are so common in programming, C++ offers a special set of operators designed specifically for these jobs. Table 3-9 shows the combined assignment operators, also known as compound operators or arithmetic assignment operators.

**Operator Example Usage Equivalent To**

+= x += 5; x=x+5;

-= y -= 2; y=y-2;

\*= z \*= 10; z = z \* 10;

/= a /= b; a = a / b;

%= c %=3; c = c %3;

**Source code samples featured:**

Assignment\_types.cpp