1.12 Succeeding with Shortcuts

## Succeeding with Shortcuts

Are any of the following algebraically valid?

n = n + 1

x = x \* z

t = t + .039

a = a - 1

b = b / 15

d = d - q

Substitute numbers for each of the variables and see if the quantities on either side of the equal sign are the same. For example, is 2 = 2 + 1 valid?

What may be surprising is that these arithmetic expressions are perfectly valid in Java because here, = is not the same equal sign you you know from math class! It is used for assignment.

Always verify your answers. Even if a program produces results, it may not be correct. There can be unexpected surprises. In this lesson, you will see that numbers can be misrepresented in memory.

### **Part 1**

In programming, the equal sign (=) is referred to as the assignment operator, and it is interpreted differently than in math.

When the computer encounters a line of Java code containing an assignment operator (=), execution proceeds from right to left in a two-step process.

* First, the answer to the expression on the right side of the equal sign is calculated using the values assigned to the variables.
* After the answer is computed, it is assigned to the variable on the left of the equal sign.

In other words, the answer calculated on the right side of the assignment operator replaces the value assigned to the variable on the left side of the equal sign. That's why each of the following would work in Java, but not in algebra.

n = n + 1

x = x \* z

t = t + .039

a = a - 1

b = b / 15

d = d - q

In the example n = n + 1, the expression n + 1 would be solved first. If the current value of n is 9, then 9 is added to 1 to get a result of 10. The resulting value 10 would then be assigned to n. The variable n now has a value of 10.

While such arithmetic expressions may seem odd at first, they are the key to the processing power of computers.

Computer programming is a very exacting discipline. The compiler will not ignore syntax violations or runtime errors, so it seems a little strange that the memory doesn't always represent decimal numbers accurately! Discover some of the pitfalls and surprises related to Java arithmetic along with some programming shortcuts.

* Create a new project called 01.12 Arithmetic Expressions in the Mod01 Lessons folder.
* Download the [AdmissionsV1.java](https://lti.flvsgl.com/flvs-cat-content/5bbomn7ui48lhcaijvtrp5l76d/flvs-cat-session/apcomputersciencea_v20/module01/lesson12/docs/02_07b/admissionv1.java) file and save to the newly-created folder.
* Open the [01.12 Desk Check: AdmissionsV1](https://lti.flvsgl.com/flvs-cat-content/5bbomn7ui48lhcaijvtrp5l76d/flvs-cat-session/apcomputersciencea_v20/module01/lesson12/pop/02_07b/02_07b_popup01.htm).

This assignment will give you practice with two useful coding shortcuts: the arithmetic/assignment operator and the increment operator. Most programmers like to write code, but they don't like to type, so shortcuts save time. Plus, they just look neat!