

## Lesson 4.....Using Numeric Variables

### The assignment operator:

The assignment operator is the standard equal sign (=) and is used to “assign” a value to a variable.

```
int i = 3; // Ok,...assign the value 3 to i. Notice the direction of data flow.
```

```
3 = i; // Illegal! Data never flows this way!
```

```
double p;  
double j = 47.2;  
p = j; // assign the value of j to p. Both p and j are now equal to 47.2
```

### Multiple declarations:

It is possible to declare several variables on one line:

```
double d, mud, puma; //the variables are only declared  
double x = 31.2, m = 37.09, zu, p = 43.917; //x, m, & p declared and initialized  
// zu is just declared
```

### Fundamental arithmetic operations:

The basic arithmetic operation are +, -, \* (multiplication), / (division), and % (modulus).

Modulus is the strange one. For example, `System.out.println(5%3);` will print 2. This is because when 5 is divided by 3, the **remainder** is 2. **Modulus gives the remainder.** Modulus also handles negatives. The answer to  $a\%b$  always has the same sign as  $a$ . The sign of  $b$  is ignored.

### PEMDAS:

The algebra rule, PEMDAS, applies to computer computations as well. (PEMDAS stands for the order in which numeric operations are done. P = parenthesis, E = exponents, M = multiply, D = divide, A = add, S = subtract. Actually, M and D have equal precedence, as do A and S. For equal precedence operation, proceed from left to right. A mnemonic for PEMDAS is, “Please excuse my dear Aunt Sally”... See [Appendix H](#) for the precedence of all operators.)

```
System.out.println(5 + 3 * 4 - 7); //10  
System.out.println(8 - 5*6 / 3 + (5 - 6) * 3); //-5
```

### Not the same as in Algebra:

An unusual assignment....consider the following:

```
count = count + 3; //this is illegal in algebra; however, in computer science it  
//means the new count equals the old count + 3.
```

```
int count = 15;  
count = count + 3;
```

```
System.out.println(count); //18
```

### Increment and Decrement:

The increment operator is ++, and it means to add one. The decrement operator is --, and it means to subtract one:

x++;	means the same as	x = x + 1;
x--;	means the same as	x = x - 1;
x++	is the same as ++x	(the ++ can be on <b>either</b> side of x)
x--	is the same as --x	(the -- can be on <b>either</b> side of x)

```
int y = 3;
y++;
System.out.println(y); //4
```

### Compound operators:

	<u>Syntax Example</u>	<u>Simplified meaning</u>
a.	+= x += 3;    →	x = x + 3;
b.	-= x -= y - 2;    →	x = x - (y - 2);
c.	*= z *= 46;    →	z = z * 46;
d.	/= p /= x-z;    →	p = p / (x-z);
e.	%= j %= 2    →	j = j%2;

### Code Examples

```
int g = 409;
g += 5;
System.out.println(g); //414
```

```
double d = 20.3;
double m = 10.0;
m *= d - 1;
System.out.println(m); //193.0
```

### The whole truth:

Actually, the full truth was not told above concerning x++. It does not always have the same effect as does ++x. Likewise, x-- does not always have the same effect as does --x.

x++ increments x **after** it is used in the statement.  
 ++x increments x **before** it is used in the statement.

Similarly,

$x--$  decrements  $x$  **after** it is used in the statement.  
 $--x$  decrements  $x$  **before** it is used in the statement.

### Code Examples

```
int q = 78;
int p = 2 + q++;
System.out.println("p = " + p + ", q = " + q); //p = 80, q = 79
```

```
int q = 78;
int p = ++q + 2;
System.out.println("p = " + p + ", q = " + q); //p = 81, q = 79
```

### **Integer division truncation:**

When dividing two integers, the fractional part is truncated (thrown away) as illustrated by the following:

```
int x = 5;
int y = 2;
System.out.println(x / y); //Both x and y are integers so the "real" answer of 2.5
//has the fractional part thrown away to give 2
```

## **Exercise on Lesson 4**

Unless otherwise directed in the following problems, state what is printed. Some of these problems may have incorrect syntax and in those cases you should answer that the code would not compile.

1. `int h = 103;`  
`int p = 5;`  
`System.out.println(++h + p);`  
`System.out.println(h);`
2. Give three code examples of how to increment the integer  $j$  by 1.
3. `double def;`  
`double f = 1992.37;`  
`def = f;`  
`System.out.println(def);`

4. Write a **single** line of code that will print the integer variable *zulu* and **then** decrement its value by 1.
  
5. 

```
int a = 100;
int b = 200;
b/=a;
System.out.println(b + 1);
```
  
6. Write a **single** line of code that uses the compound operator, `-=`, to subtract  $p-30$  from the integer value *v* and store the result back in *v*.
  
7. Write a single line of code that does the same thing as #6 but without using `-=`.
  
8. 

```
int p = 40;
int q = 4;
System.out.println(2 + 8 * q / 2 - p);
```
  
9. 

```
int sd = 12;
int x = 4;
System.out.println( sd%(++x) );
System.out.println(x);
```
  
10. 

```
int g;
3 = g;
System.out.println(++g*79);
```

What is the result?
  
11. On a single line of code declare *m*, *b*, and *f* to be *double* and on that same line initialize them all to be 3.14.
  
12. On a single line of code declare *x*, *y*, and *z* all to be of integer type.
  
13. 

```
int m = 36;
int j = 5;
m = m / j; // new m is old m divided by j
System.out.println(m);
```

What's printed?
  
14. 

```
System.out.println(3/4 + 5*2/33 -3 +8*3);
```

What's printed?

15. What is the assignment operator?
16. Write a statement that stores the remainder of dividing the variable  $i$  by  $j$  in a variable named  $k$ .
17. `int j = 2;`  
`System.out.println(7%3 + j++ + (j - 2) );`
18. Show three different ways to decrement the variable  $j$ .

## Project... Cheating on Your Arithmetic Assignment

Create a new project called *ArithmeticAssignment* with a class called *Tester* that will calculate and print the results of the following arithmetic problems:

$$79 + 3 * (4 + 82 - 68) - 7 + 19$$

$$(179 + 21 + 10) / 7 + 181$$

$$10389 * 56 * 11 + 2246$$

The printout should look like the following:

$$79 + 3 * (4 + 82 - 68) - 7 + 19 = 145$$

$$(179 + 21 + 10) / 7 + 181 = 211$$

$$10389 * 56 * 11 + 2246 = 6401870$$