Getting Started With Java

Mathematical Operations

Basic Math in Java

operation	operator	example
add	+	2 + 3 = 5
subtract	-	2 – 3 = -1
multiply	*	2 * 3 = 6
divide	/	6 / 2 = 3

Order of Operations

Recall: BEDMAS

- B = Brackets
- E = Exponents
- D = Division
- M = Multiplication
- A = Addition
- S = Subtraction

Order of Operations

Consider the following examples. The order of the numbers and operations are all the same, but the placement of brackets gives very different results.

```
5.0 * 4.0 + 3.0 / 2.0 = 21.5

5.0 * (4.0 + 3.0) / 2.0 = 17.5

5.0 * (4.0 + 3.0 / 2.0) = 27.5

(5.0 * 4.0 + 3.0) / 2.0 = 11.5
```

Declaring Numeric Variables

- reserves space in memory for the variable
- specify the type of data to be stored
- name using camelCase

```
int count;
double average;
```

Assigning Values to Variables

A variable is used to store information. To set or change the value in a variable, Java uses a single equals sign (=).

(1) Perform all calculations on the right side of the equals sign.

(2) Store the value in the variable on the left side of the equals sign.

Assigning Values to Variables

What you see:

```
int count = 1;          // count is now 1
count = count + 2;      // count is now 3
count = count + 3;      // count is now 6
```

What Java sees:

```
int count = 1;          // count is now 1
count = 1 + 2;          // count is now 3
count = 3 + 3;          // count is now 6
```

Modulo Operator %

A specialized, but very useful, operator that determines the remainder from integer division (no decimals, only fractions).

e.g., For 7 divided by 3, we see that 3 will fit into 7 twice. Unfortunately, two 3's only makes 6. We are still one short of 7, and this is the remainder.

In Java, 7 % 3 is equal to 1.

In general, a % b is asking: "For a divided by b, what is the remainder?"