AP Computer Science   Name: ______________________________
Chapter 6 Notes (4)

DATA TYPES, VARIABLES, and ARTHMETIC

<u>**Numeric Type Conversions**</u>:

Can you perform binary operations with two operands with two operands of different types? Yes. If an integer and a floating-point number are involved in a binary operation, Java automatically converts the integer to a floating-point value. So,

`3 * 4.5` is the same as `3.0 * 4.5`

You can also assign a value to a numeric variable whose type supports a larger range of values; thus, for instance, you can assign a `int` value to a `double` variable. You cannot, however, assign a value to a variable of a type with a smaller range unless you use *type casting*. *Casting* is an operation that converts a value of one data type into a value of another data type. Casting a type with a small range to a type with a larger range is known as *widening* a type. Casting a type with a larger range to a type with a smaller range is known as *narrowing* a type. Java will automatically widen a type, but you must narrow a type explicitly.

The syntax for casting a type is to specify the target type in parentheses, followed by the variable's name or the value to be cast.

Examples:

`System.out.println(1 / 2);`	Displays	0
`System.out.println(1.0 / 2);`	Displays	0.5
`System.out.println(1 / 2.0);`	Displays	0.5
`System.out.println(1.0 / 2.0);`	Displays	0.5
`System.out.println((double)1 / 2);`	Displays	0.5
`System.out.println((int)1.7);`	Displays	1

<u>**Note**</u>:

In the previous lessons we learned that the compound operator expression `j += x;` was equivalent to `j = j + x;`. Actually, for **all compound operators** there is also an **implied cast** to the type of `j`. For example, if `j` is of type `int`, the real meaning of `j += x;` is:

```
j = (int)(j + x);
```