Arithmetic Operators

C++ Notes Mrs. Alano

Five simple arithmetic operators:

- the addition operator (+)
- the subtraction operator (–)
- the multiplication operator (*)
- the division operator (/)
- the modulus operator (%)

Addition, subtraction, multiplication, division, or modulus of any two integers results in an integer

- For example, 7 / 3 evaluates to 2

- Mixed expression: operands have different data types
  - For example, 3.2 * 2
- Unifying type:
  - Data type of the value that occupies more memory
  - All types in the expression are temporarily converted to a unifying type
- The order of precedence of unifying types from highest to lowest
  - long double
  - double
  - float
  - unsigned long
  - long
  - unsigned int
  - int
  - short
  - char
- Cast: transform a value to another data type
- Implicit cast: automatic cast, or transformation, that occurs when you assign a value of one type to a type with higher precedence
  - int answer = 2.0 * 7;
- Explicit cast: deliberate cast
  - intResult = (int)doubleVariable;

- **Modulus (%) gives the remainder of integer division**
  - 7 % 3 results in 1
  - -10 % 8 produces -2
  - -10 % -8 produces -2
- Can be used only with integers
- Can be used to extract digits from numbers
  - 6,543 % 10 is 3
  - 6,789 % 10 is 9

Compound Assignment Operators:
C++ has convenient shortcuts for combining arithmetic operations with assignment. The following table summarizes the ***compound assignment*** operators:

Compound Assignment:	Is the same as:
`a += b;`	`a = a + b;`
`a -= b;`	`a = a - b;`
`a *= b;`	`a = a * b;`
`a /= b;`	`a = a / b;`

Example:

The following statement:

```
amt = amt + taxAmt;
```

Can be rewritten as:

```
amt += taxAmount;
```

Increment/Decrement Operators:
C++ has special **increment/decrement** operators (one of which gave the language its name). These operators are used as shorthand for incrementing and decrementing an integer value:

Increment/Decrement:	Is the same as:
`a++;`	`a = a + 1;`
`a--;`	`a = a - 1;`
`++a;`	`a = a + 1;`
`--a;`	`a = a - 1;`

See operatorsex.cpp