Math operators Notes

Tutorial 9, Session 9.3 Carey 6e
JavaScript operations are similar to operations in other languages such as C++.

1. **Mathematical operators**
the usual operators are + (add), - (subtract), * (multiply) and / (divide). Also ++x or x++ which increments the value of x by 1.

a. **Follow the order of operations** as we learned in arithmetic. Recall PEMDAS (Please Excuse My Dear Aunt Sally with Parentheses performed first, then exponents, from left to right multiplication and division, from left to right addition and subtraction.) (Note modulus % follows after division.)

try: 2 + 3 * 8 (answer: 26)

b. **Division** is like arithmetic division and returns decimal answers. Ex: 10/3 = 3.3333. (This contrast to the result in the language C++, where 10/3 truncates to 3.)

try: 9/2 (answer = 4.5)

c. **x++ and ++x** (**incrementing** the value of x by 1)
FYI: You will not be required to know the difference between x++ and ++x for Test 3.

1) In x++, first the current value of x is used in the execution of the statement in which x++ resides and *then* x is incremented by 1.
2) In ++x, first the value of x is incremented by 1, and *then* this new value of x is used in the execution of the statement in which ++x resides.
3) Example:

```
x = 2;
document.write(x++);
document.write(x + "<br />");
renders:
2
3
```

4) Example:

```
x = 2;
document.write(++x);
document.write(x + "<br />");
renders:
3
3
```

2. **Assignment operator abbreviations**
Example: x += 5 is an abbreviation of x=x+5;

Example: trace the code. What is final value of x?

```
x = 2;
x += 5;
```

The final value of x is 7. Here are the steps:

```
x = 2;
x = x + 5;      <-- Rewrite x += 5; as its equivalent: x = x + 5;
x = 2 + 5;      <-- The righthand side expression is done first. Since x was assigned 2 in a previous
                    step, replace x with 2 to get 2 + 5 or 7.
x = 7;          <-- 7 is assigned to x.
```

Other abbreviations are -=, *= and /=