Model 1 Order of Operations

Java follows a specific order for math and other operations. For example, multiplication and division take ***precedence*** over addition and subtraction. The following table lists several Java operators from highest precedence to lowest precedence.

Parenthesis	`( )`
Unary (positive or negative signs)	`+ -`
Multiplicative	`* / %`
Additive	`+ -`
Assignment	`=`

For the following questions, assume you have these two variables:

```
int x;
double y;
```

Questions (10 min) Start time:

1. What operator has the lowest precedence? Why do you think Java is designed that way?

Assignment has the lowest precedence so that all other operations happen first (before the final value is stored in memory).

2. What operator has the highest precedence? Why do you think Java is designed that way?

Parenthesis have the highest precedence so that you can impose a specific order. For example, `2 * (3 + 4)` will perform addition before multiplication.

3. The `+` and `-` operators show up twice in the table of operator precedence. For the Java statement `x = 5 * -3;` explain how you know whether the `-` operator is being used as an ***unary*** or ***binary*** operator.

It matters what is to the left or right of the operator. In this example, the `-` is preceded by a `*`, so it must be unary.

4. Determine the order of operations in the statement: `x = 5 * -3;`

a) First operator to be evaluated: -

b) Second operator: *

c) Third operator: =

5. Determine the order of operations in the statement: `y = 9 / 2;`

a) First operator to be evaluated: /

b) Second operator: =

6. Based on your answer to the previous question, explain why the variable `y` would be assigned 4.0 (as opposed to 4 or 4.5).

Since `y` is a floating-point variable, the integer result 4 would be stored as 4.0 in memory.

7. Rewrite the assignment for `y` so that it would be set correctly to 4.5. (Hint: you'll need to recall what you learned about division in **??**.)

```
y = 9.0 / 2.0;
```