Arithmetic Expressions in Java

Objectives

At the end of this exercise, students will:

- Be able to evaluate the binary arithmetic expressions in Java.
- Be able to create arithmetic expressions in Java to solve arithmetically based problems.

Roles

- Coordinator – will make sure the team stays on task
- Presenter – will only be used if there is a question about your group's responses
- Recorder – will write on the board the group answers to BOARD questions
- Reflector – if present will fill out the exit pass, otherwise, team members will do so

Getting ready

1. All team members - record your notes on your own response sheet.

Background – Relearning arithmetic – Channeling your inner 4th grader

While most of the Java operations act as we expect them to from Math classes, integer division id different. This part of the exercise will help you to practice how integer division works. It is very similar to the way you first learned division.

1. Each bag should have 19 pennies. Divide the pennies so that each person in the group gets the same number of pennies.

   $n \overline{)\ 19}$

   (n is the number of people in your group)

2. You should have (n + 1) stacks of pennies, one for each person in the group and one that has the leftovers. Since you can't divide a penny, we say that we have a quotient (the number of even size stacks) and a remainder (the stack that contains the rest).

3. How many people in your group today? _____

4. What is the quotient (the number of pennies in the even stacks)? _____

5. How many pennies are left over? _____

In Java, integer division is like this kind of division. 12/7 is 1 because that is how many times we can put 7 into 12. 12 % 7 is 5 since we can put 7 into 12 once, but then we have 5 extras.

Let's try a few more with the pennies. Count out 13 pennies.

$$4\overline{)\,13}$$

1. If you start with 13 pennies and divide them 4 ways (into 4 groups) how many pennies in each group? (13 / 4) _____

2. How many pennies are left over? (13 % 4) _____

Now a harder one.

Count out 4 pennies. Divide those 4 pennies into 13 even piles.

$$13\overline{)\,4}$$

1. How many pennies do you have in each of the 13 piles? 4 / 13 _____

2. How many are leftover? 4 % 13 _____

Understanding Integer Division – more practice

1. What is the result of the each of the following integer division operations?

8 / 3	
8 % 3	
127 / 22	
127 % 22	
22 / 127	
22 % 127	

2. **BOARD** – On the board, write the result of the expression, 5 % 6 / 3. (The expression is evaluated from left to right.)

3. Describe the process of performing integer division. How did you figure out the division and modulus results from question 1?

More Arithmetic operations in Java – Gaddis 2.5

+	Addition	Binary	total = cost + tax;
-	Subtraction	Binary	cost = total – tax;
*****	Multiplication	Binary	tax = cost * rate;
/	Division	Binary	salePrice = original / 2;
%	Modulus	Binary	remainder = value % 5;

This table displays the 5 Java arithmetic operations in Java. They are **binary**, meaning that they take two operands. saw addition in the lab already seen addition. Notice in the examples that every operator is separated from the operands by a single blank character. This is to provide better readability of the operations for the human reader. The compiler ignores any form of whitespace including spaces, tabs, and new line characters.

The order of precedence of these operations is * / %, then + -. Within levels, operations are carried out from left to right.

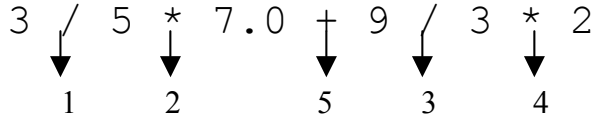
Also, something you need to know. Java operations must be done on the same data type. So if we mix integers and doubles, the overall operation will be carried out with double numbers. The integer value will widen to a double and the operation carried out. This happens operation by operation.

The model

The following are examples of arithmetic expressions in Java.

Expression	Data Type of Result	Value of Result
3 + 5	integer	8
5 – 7.0	double	-2.0
5.0 * 1.0	double	5.0
10.0 / 2.0	double	5.0
10 / 3	integer	3
10 % 3	integer	1
7 % 6	integer	1
7 / 6	integer	1
6 / 7	integer	0
6 % 7	integer	6

The following is an example of how Java will calculate an expression with mixed operations and operands. The arrows and numbers below show in what order Java will carry out the evaluation of this expression.



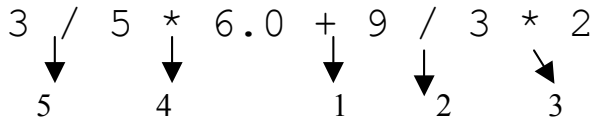
Exploring the model

Using the chart of examples, answer the following: **(YOU MAY NOT USE A CALCULATOR)** Assume that the result of each operation will be used in later steps, for example, step one result is used in step 2, etc.

1. What is the result of operation 1?
2. What is the data type of operation 1?
3. What is the result and data type of operation 2?
4. What is the result and data type of operation 3?
5. What is the result and data type of operation 4?
6. What is the result and data type of operation 5?
7. **BOARD** – Put your team name on the board and your answer to number 6.

Extending the model

1. Just like in math class, parentheses are used in Java to force an operation to be evaluated before operations outside of the parentheses are evaluated.
2. In the complex expression above, put in parentheses to document the order of operations as described by the numbers.
3. In the complex expression below, put in parentheses to force Java to evaluate the expression based in the order shown beneath the arrows.



4. **BOARD** Evaluate the expression from item number 3. Put your answer on the board under your group name.

Applying arithmetic operations to problems

1. What expression(s) would you use to solve this problem? Miles per gallon(mpg) is calculated by taking the number of miles driven divided by the number of gallons of gasoline purchased. If we drove 100 miles and purchased 3.2 gallons of gasoline, what is the mpg?
2. **BOARD** What expression(s) would you use to solve this problem? Calculate the number of minutes, seconds, and hours in 7000 seconds?
3. Write another example of turning seconds into the equivalent hours, minutes and seconds.

Now a program – use a separate piece of paper

1. Write the body of a program that will calculate the mpg for 1000 miles and 7 gallons of gas. You only need to include the statements that would go inside the `main` block. You should include a meaningful output statement.

```
public static void main (String args[])
{
}
```

Coordinator – Have someone from the team erase your team's responses from the group responses (one individual sheet or one piece of paper) and the exit pass.