Lab - Arithmetic Operators

Tutorial Lab 1: Arithmetic Operators

Arithmetic operations in Java are mostly the same as what you learned in math class. However, the symbols used in Java can be different.

Operation	Symbol	Notes
Addition	+	
Subtraction	-	
Multiplication	*	
Division	/	Always returns a float
Modulo	%	Returns the remainder after division is performed

Use the text editor open in the left pane, and enter the following code:

```
System.out.println(10 + 3);
System.out.println(10 - 3);
System.out.println(10 * 3);
System.out.println(10 / 3);
System.out.println(10 % 3);
```

1. Addition works as expected.
2. Subtraction works as expected.
3. Multiplication works as expected.
4. Division with integers will return a truncated integer result.
5. Modulo returns `1` because that is the remainder (not the decimal) after division is performed.

Lab - Strings

Tutorial Lab 2: Strings

You can use the `+` and `*` operators with strings, even though the result is not based on math. Using the `+` operator with strings is called concatenation.

Use the text editor open in the left pane, and enter the following code:

```
String string_1 = "hip ";
String string_2 = string_1 + string_1;
String string_3 = "hoo";
String string_4 = "ray!";
String string_5 = string_3 + string_4;
System.out.print(string_2);
System.out.println(string_5);
```

1. Assign the value `"hip "` to the variable `string_1`. Note the inclusion of a space after the word `hip`.
2. The variable `string_2` will have the value `"hip hip "` because `string_1 + string_1` repeats the value of `string_1` two times.
3. Declare `string_3` and assign it the value `hoo`.
4. Declare `string_4` and assign it the value `ray!`.
5. Declare `string_5` and assign it the value of `string_3` combined with the value of `string_4` (`hooray!`).
6. Print the value of `string_2` (`hip hip`) without the newline character.
7. Print the value of `string_5` (`hooray!`) to the end of `string_2`.

Lab - Order of Operators

Tutorial Lab 3: Order of Operations

Java uses PEMDAS when determining the order of operations.

P Parentheses
E Exponents - powers & square roots
MD Multiplication & Division - left to right
AS Addition & Subtraction - left to right

PEMDAS

▼ **Modulo and PEMDAS**
Since modulo is based on division, modulo operations happen at the time of multiplication and division, going from left to right.

Use the text editor open in the left pane, and enter the following code:

```
System.out.println(5 * 8 / 3 + (18 - 8) % 2 - 25);
```

Below are the steps that Java takes when evaluating the code above.
1) `5 * 8 / 3 + (18 - 8) % 2 - 25`
1) `5 * 8 / 3 + 10 % 2 - 25`
1) `40 / 3 + 10 % 2 - 25`
1) `13 + 10 % 2 - 25`
1) `13 + 0 - 25`
1) `13 - 25`
1) `-12`

Lab - Boolean Operators

Tutorial Lab 4: Boolean Operators

Boolean operators are those operators which will return either true or false.

Operation	Symbol	Notes
Equal to	`==`	The = operator is assignment operator, not the equality operator
Not equal to	`!=`	
Less than	`<`	
Less than or equal to	`<=`	
Greater than	`>`	
Greater than or equal to	`>=`	
And	`&&`	Compares two boolean expressions. Both must be true for the whole to be true. Everything else is false.
Or	`||`	Compares two boolean expressions. Both must be false for the whole to be false. Everything else is true.
Not	`!`	Returns the opposite of a boolean expression.

Use the text editor open in the left pane, and enter the following code:

```
System.out.println((5 > 7) && (false || 1 < 9) || 4 != 5 && ! (2
        >= 3));
```

Below are the steps that Java talks when evaluating the code above.

Evaluate all arithmetic operators according to PEMDAS

1. `(5 > 7) && (false || 1 < 9) || 4 != 5 && ! (2 >= 3)`
2. **`false`** `&& (false || 1 < 9) || 4 != 5 && ! (2 >= 3)`
3. `false && (false || ` **`true`**`) || 4 != 5 && ! (2 >= 3)`
4. `false && (false || true) || ` **`true`** `&& ! (2 >= 3)`
5. `false && (false || true) || true && ! ` **`false`**

Evaluate all boolean operators (order: parenthesis, not, and, or)

6. `false && ` **`true`** `|| true && ! false`
7. `false && true || true && ` **`true`**
8. **`false`** `|| true && true`
9. `false || ` **`true`**
10. **`true`**

Lab Challenge - Operators

Operators Challenge

Write a boolean expression that incorporates one of the equality operators, one of the less than operators, one of the greater than operators, and two of the logical operators. The result of your boolean expression must be `false`.

Equality	Less Than	Greater Than	Logical
==	<	>	&&
!=	<=	>=	||
			!