

Lab 03: Expression Evaluation

Release date: Week of September 5th, 2016

Due date: By the end of your lab session on the same week

Goals

- ★ To learn using Strings for data representation
- ★ To learn methods to convert from String Objects to numeric data types

Prerequisites

- Knowledge of String functions
- Knowledge of methods that convert from String Objects to numeric data types
- Knowledge of operators precedence
- Knowledge of some Math library functions

Main Task

You are required to write a program that reads a sequence of characters (String) that represent an arithmetic expression. Your program should evaluate this arithmetic expression and present the answer to the user. The expression has the following fixed format.

$$(-n_1 + \sqrt{n_2 * n_3 - 4 * n_4 * n_5}) / (2 * n_6)$$

Where n_i is an integer positive number; i is an integer number from 1 to 6.

The following are examples of possible input expressions:

- 1) $(-10 + \sqrt{25 * 4 - 4 * 3 * 2}) / (2 * 7)$
- 2) $(-5 + \sqrt{100 * 6 - 4 * 12 * 4}) / (2 * 123)$

Your task is to parse the input expression, extract the values of n_1 through n_6 and evaluate the expression considering the operators precedence rules. For this lab, you are not required to do any input checking or error handling. It is thus important to make sure that any input expression you try will evaluate smoothly without generating errors; you should make sure that the input String is in the correct format, that the value that you get the square root for $(n_2 * n_3 - 4 * n_4 * n_5)$ is a positive number and that n_6 is not a zero.

In order to solve this problem, your program should proceed in the following steps:

1. Get the expression to evaluate as input using the Scanner class methods
2. Extract the values of n_1 through n_6 from the input expression
3. Evaluate the expression based on values found in Step 2.

Observe that the position of the numbers relative to the operators does not change, e.g., there will always be a ‘*’ between n_2 and n_3 . This will help you in performing Step 2. After extracting all the numbers, you can then write the formula corresponding to the original expression using mathematical operators in Step 3.

All your code should reside in the main method of a class called “ExpressionEvaluator”. The location of your Java file “ExpressionEvaluator.java” should be under the directory “Desktop/cs180/lab03”.

Turning-in Your Work

After you are done with your work, run your program several times on different input expressions. Make sure that the output is correct. If everything works properly, call your GTA to evaluate your work and assign your grade.

In addition to being evaluated by a GTA, you are required to turn-in your code. Use the following command for turning-in your code.

```
$ turnin -c cs180=COMMON -p lab03 lab03
```

Rubric

This lab is worth 50 points. You must receive your grade **before** you leave the lab session. If you are not done with all the work, you may get partial credit. Therefore, you should save all your data properly and make sure that your work is evaluated by the GTA at the end of the lab session.

Good Luck