**Programming Assignment # 3**

**Objective:** To use stacks to solve a programming problem

Write a Java program to evaluate infix expression (not necessarily fully parenthesized) using stacks. The priority of operators shall be taken into account. See <https://en.wikipedia.org/wiki/Infix_notation> for an example of an infix notation.

**Tip:** You would probably need two stacks (a stack for numbers and stack for characters).

**Input:** The program should ask the user to input the infix expression

**Output:** The result of the calculation

**Assumptions:**

* You can use the Stack implementation of the Java API, or use one of the ones we have covered in class.
* Assume that the input expressions contain only positive integer values, and that the output is always an integer.
* The input that the user types can contain any string with the following characters: ( , ) , numbers operators, where numbers is any positive integer and operators are any of +, -, \* , /
* Your program assumes that the expression is well-formed, but must check is the parenthesis are balanced. If parenthesis are not balanced, then the program stops and tell the user that there was an error (you may want to use a stack for this as well).

**Deliverables:** .zip file with your IntelliJ project files. Name your .zip file as lastname\_fi.zip where fi stands for the initial letter of your first name (e.g, perez\_a.zip).

**Example of execution:**

**Example # 1**

**\*\*\*\*Welcome to infix evaluator \*\*\*\***

**Type an infix expression: (4+5\*6)/2**

**Parenthesis are balanced**

**The result of the expression is: 17**

**\*\*\*\*Program ends\*\*\*\*\*\***

**Example # 2**

**\*\*\*\*Welcome to infix evaluator \*\*\*\***

**Type an infix expression: (4+5\*6/2**

**Parenthesis are not balanced. Run and type again your infix expression.**

**\*\*\*\*Program ends\*\*\*\*\*\***