**Wentworth Institute of Technology**

**Department of Computer Science and Networking.**

**Comp 2000 Data Structures**

**Lab #2 Stack applications** (total 100 pts)

*A preparation for the programming project “Stacks”*

Part1. (20 pts)

Finish both LinkedStack and ArrayStack implementations.

------------------------------------------------------------------------------------------------------------

Part 2. (80 pts)

In the file 04b\_StackApplications.docx from the directory “Lectures” the algorithms for infix-to-postfix converter and postfix expression evaluation are described.

Translate those algorithms in Java code, creating two classes: InfixToPostfix and PostfixEvaluator.

The method *InfixToPostfix.convert(String infix)* creates the new string; the method *PostfixEvaluator.eval (String postfix)* returns the computed value.

You can see the code for *InfixToPostfix.convert* and *InfixToPostfix.checkBalance* (under different names)in the Java file supplied in the lectures. The method *PostfixEvaluator.eval*  is your responsibility.

For testing use the LinkedStack class from Part1.

Assume that the string for evaluation contains only one-digit numbers.

The result of the evaluation is supposed to be a floating-point value.

Test the method by asking user to enter a string.

Upon user input, you should check whether the string is parentheses-balanced.

Here is a possible output:

|  |
| --- |
| Enter a string for evaluation, digits only: 1+3^3^2/5  The result: 3937.6 |

|  |
| --- |
| Enter a string for evaluation, digits only:  (7-2)+(4\*9  Input error: unbalanced string |

Make sure the code works if you replace LinkedStack with ArrayStack.

Extra credit (10 pts): enhance code so that the methods work with multi-digit numbers.