

# CS 2401 Assignment #9

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

## CS 2401 – Elementary data structures and algorithms

### Fall 2022: Lab 9

**Due Date: Monday, November 15, 2022 – end of the day.**

**Objective:** This lab assignment is designed to improve skills of using **stacks** for problem solving.

**Assignment:** We are given the Stack class (called Stack.java). This class uses a Node class; which is provided in the Node.java file. **You will only write in the StackRunner.java file.** Do NOT change the Stack.java and Node.java files.

In this assignment you will solve two problems using stacks. **You can only use the stack with the Stack.java file provided with this assignment for stack operations; NO Stack classes from any external packages are allowed.** The Stack.java file contains the code that we discussed in the class.

**You are not allowed to use any java.util container data structures for any assignment.** For example, the use of java Vector, ArrayList, Stack, LinkedList, Set classes is strictly prohibited.

It is alright to use java.io classes. It is alright to use java.util.StringTokenizer or something similar if required to tokenize a string.

#### **Problem 1: Check if an expression is balanced.**

**Task:** Complete the method named **isBalanced** in **StackRunner.java**. Do NOT change the header of the **isBalanced** method.

Check if an expression is balanced with parentheses. Given a string containing (possibly) an infix expression, use a stack to determine if the string contains a parentheses-balanced expression. Work with the start '(' and closing parentheses ')' only. Ignore any other character. Please see the sample output in this document to find how the output should behave.

#### **Problem 2: Evaluate a given postfix expression.**

**Task:** Complete the method named **evaluatePostfix** in **StackRunner.java**. Do NOT change the header of the **evaluatePostfix** method.

Evaluate the postfix expression given in a String. The operators and the operands are separated by **spaces**. Only four operators are allowed +, -, \*, and /. For anything else other than a numeric operand or a valid operator, the program must inform on

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

## CS 2401 Assignment #9

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

the terminal that the postfix expression is incorrect. The program must report on the terminal for any possible error in the postfix expression. Please see the sample output of the program to realize what sort of errors there might be.

**Input:** The input is provided in a text file named `input.txt`. The input file format is explained below.

1. Any line in the input text file starting with the letter **b** will be used for problem 1 (checking balance-ness.)
2. Any line starting with the letter **p** contains a postfix expression (possibly), which should be evaluated.
3. Any line containing any other starting will be reported with the message "The line does not start with a 'b' or a 'p'. This program will skip this line."
4. Empty lines must be ignored without any message.

**NOTE:** The string passed to `isBalanced` should NOT contain the starting b. The string passed to the `evaluatePostfix` method should not contain the starting p.

The method `evaluatePostfix` should return `Double.NEGATIVE_INFINITY` in case the expression cannot be evaluated for any valid reason. Seeing the `Double.NEGATIVE_INFINITY` return value, the caller should realize that something went wrong in evaluating the postfix expression. In that case, the caller should output "The line contains an invalid postfix expression." See the sample output for an example.

A complete method `isNumber(String tk)` is provided in the `StackRunner.java` file in case you need to check if a string token is numeric. You may call the `isNumber` method from the `evaluatePostfix` method.

**You are not allowed to create any additional method.** The three methods you will work on are: the `main` method, the method named `isBalanced`, and the method named `evaluatePostfix`. **You are not allowed to change any of the headers.**

**You are not allowed to use any global variable.**

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

## CS 2401 Assignment #9

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

A sample **input.txt** file is provided below. The file is also provided with the assignment.

### Sample input:

```
b (() ( ( ) ) ) ( ( ) )
b ( ) ) ( ) ( ( ) )
b (i-j)+3(10-8)-2(3+1)
b ( ( ) ) ( ( ( ) ( )
b ( ( ) ( ) ) ) ( )
( ( ) ( ) ) ) ( )

p 10 2 / 5 *
p 5 10 - 3 9 - * 5 4 - 4 2 / + / 2 27 + 300 3 / / -

p t b *
p 15 8 / +
p 15 8 4 / +
p 5 10 / 11
```

The **main** method, **isBalanced**, and **evaluatePostfix** methods must be written in such a way that the sample output of your program matches the following sample output for the given **input.txt** file.

### Sample output

```
The line is: b (() ( ( ) ) ) ( ( ) )
Checking the balance of the expression.
The expression: (() ( ( ) ) ) ( ( ) )
The expression is: balanced.
-----
The line is: b ( ) ) ( ) ( ( ) )
Checking the balance of the expression.
The expression: ( ) ) ( ) ( ( ) )
The expression is: imbalanced.
-----
The line is: b(i-j)+3(10-8)-2(3+1)
Checking the balance of the expression.
The expression: (i-j)+3(10-8)-2(3+1)
The expression is: balanced.
-----
The line is: b ( ( ) ) ( ( ( ) ( )
Checking the balance of the expression.
The expression: ( ( ) ) ( ( ( ) ( )
The expression is: imbalanced.
-----
The line is: b ( ( ) ( ) ) ) ( )
Checking the balance of the expression.
The expression: ( ( ) ( ) ) ) ( )
The expression is: imbalanced.
-----
```

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

## CS 2401 Assignment #9

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

The line is: (()())()

The line does not start with a 'b' or a 'p'. This program will skip this line.

-----  
The line is: p 10 2 / 5 \*

The evaluation of this postfix expression is: 25.0

-----  
The line is: p 5 10 - 3 9 - \* 5 4 - 4 2 / + / 2 27 + 300 3 / / -

The evaluation of this postfix expression is: 9.71

-----  
The line is: p t b \*

The operator t is not recognizable.

The line contains an invalid postfix expression.

-----  
The line is: p 15 8 / +

Trying to pop when stack is empty

Not enough operand in stack to apply the operator +

The line contains an invalid postfix expression.

-----  
The line is: p 15 8 4 / +

The evaluation of this postfix expression is: 17.0

-----  
The line is: p 5 10 / 11

The postfix expression has extra elements.

The line contains an invalid postfix expression.

### Deliverables:

You will submit **only the updated StackRunner.java file.**

### Grading Criteria:

The grading of this assignment is based on correctness, functionality, output, and readability/documentation of the program. A correct program should have exactly the same output as the sample output for the given input.txt file. The input.txt file will be changed during grading to validate correctness further.

- [20 points] The Program **compiles and runs**.
- [10 points] The program is **indented** and **documented** properly.
- [10 points] The program uses the correct **variable types** and **names**.
- [60 points] The methods are implemented correctly.

■ **Late submission: [-10] points for every 24 hours after the deadline.**

If you need any clarification, please ask your TA for further details.

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.