ENSF 594 – Principles of Software Development II

Summer 2021

Lab Assignment # 4

Application of Stack

Total Marks: 25

Due Date: Tuesday, Aug 03, by 11:59 PM using D2L

Note:

You should use the Stack implementation. It's up to you whether you want to use Array-based or linked list based. Don't use predefined Stack available in Java. You can use class code after citing.

Q1. Brackets matching – The Delimiters on the Stack (5 marks)

Write a java program that determines whether the given set of delimiters (in our case, it will be parenthesis such as *(and)*) is validate or not.

Input:

Your program should read the input set of parenthesis from the user.

Output:

You would write on the console whether a given set of parenthesis is valid or not.

Sample Run:

Enter a String: ()
This is valid

Another run:
Enter a String:)(
This is invalid

Another run:
Enter a String: (())
This is valid

Note: For question 2 and question 3, you can assume that all the expressions contains only single digits number. There will not be any parenthesis in the expression. The operator will be +, -, * and /.

Q2. Convert from Infix to Postfix (5 marks)

We write expression with an operator (+, -, * and /) places between the operands. This is called infix notation because operator is placed between the operands. In postfix notation, operator follows the two operands. For Example A + B becaosme AB+. Below table describe Infix expression to postfix expression.

Infix	**Postfix**
A + B – C	AB+C-
A * B / C	AB*C/
A + B * C	AB*C+
A * B + C	ABC+*
A * B + C * D	AB*CD*+

Below table describe how we can convert an expression from infix to post.

Infix: A + B – C, Postfix: AB+C-

Character Read from Infix Expression	Infix expression parsed so far	Postfix Expression	Comment
A	A	A	
+	A+	A	
B	A+B	AB	
-	A+B-	AB+	When you see the -, you can copy the + to the postfix string
C	A+B-C	AB+C	
End	A+B-C	AB+C-	When you reach at the end of the expression, you can copy the -

Now consider another example

Character Read from Infix Expression	Infix expression parsed so far	Postfix Expression	Comment
A	A	A	
+	A+	A	
B	A+B	AB	
*	A+B*	AB	You cannot copy + because * has higher precedence
C	A+B*C	AB+C	When you see the C, you can copy the *
	A+B*C	ABC*	
End	A+B*C	ABC*+	When you see end of the expression,, you can copy the +

Below are the input to postfix transition rules:

Item Read from Input	**Action (infix)**
Operand	Write it to output
Operator(opThis)	If stack is empty, push opThis Otherwise, while stack not empty, repeat: Pop an item If item is an operator(opTop) and If opTop < opThis, push opTop If opTop >= opThis, output opTop Quit loop if opTOP < opThis Push opThis
No more items	While stack not empty, Pop item, output it

In the above table, < and >= symbols refers to the operator precedence. The opThis operator has just been read from the infix input, while opTop operator has just been popped off the stack

You can follow these transition rules in our first two table.

Write a java program that read an infix expression from the user (console), and convert that expression in the postfix expression.

Sample Run:

Enter infix: 2+3*4

Postfix is: 234*+

Q3. Evaluating expression (4 marks)

Below algorithm describe the evaluating postfix expression

- If item read from postfix expression is an operand, then push that item onto the stack
- If item read from postfix expression is an operator, pop the top two operands from the stack and apply the operator to them. Push the result.

When you are done, pop the stack to obtain the answer.

Sample Run:

Enter Postfix: 57+

Evaluates to 12

Submit electronically via D2L:

1. Your source code files. Your TA will run your program to verify that it works correctly. Make sure your Java program compiles and runs without issues.
2. For Q1, the main file name should be Q1_LastName_FirstName.java. Same pattern will be for Q2 and Q3.

Grading Rubric:

Functionality: produces correct output: • Produce correct output for Q1 (5 marks) • Convert infix expression to postfix (5 marks) • Evaluate the expression (4 marks)

Reference:

Some of the contents of the assignment has been adapted from Data Structures and Algorithm in Java, 2nd Edition by Robert Lafore

Other Source:

Following will be extra resources to read if you are interested.

https://introcs.cs.princeton.edu/java/43stack/

https://eecs.wsu.edu/~nroy/courses/cpts122/labs/Infix2Postfix.pdf