CALIFORNIA STATE POLYTECHNIC UNIVERSITY
Computer Science Department
CS 2400-03 ***T. Nguyen/s21***

PROJECT: **4**
DUE DATE: April 29, 2021

Description:

Implement Project 3 in chapter 25.

1. Only implement the postfix to expression tree constructor:
   +ExpressionTree(postfix:String[])
2. Write an ExpressionTest class with a main that will convert a valid postfix expression into an expression tree and use the evaluate operation to output the result from the expression.
   javac ExpressionTest 1 2 + 3 /
3. The data of each node is Double.
4. Implement a debug method in ExpressionTree that will output a postorder traversal of the nodes in the expression tree. Use this method to output the expression tree as shown in the I/O.
   +postorder():void

Required I/O:

```
Expression Tree by F. Last

Input: 1 2 + 3 /
Value: 1

Postorder Traversal:
1
2
1 : + : 2
3
+ : / : 3
```

Project report: *(PDF format)*

- Page 1: Cover page with your name, class, project, and due date
- Page 2 to 3:
  - Section 1 (**Project specification**): Your ADT description. Description of data structures used and a description of how you implement the ADT
  - Section 2 (**Testing methodology**): Description of how you test your ADT, refer to your testing output. Explain why your test cases are rigorous and complete. Demonstrate that you test each method.
  - Section 3 (**Lessons learned**): Any other information you wish to include.

Turn in:
1. Project report submitted via Blackboard.

2. There should be the following Java source files: Your Stack implementation java source files, Tree ADT, BinaryTree ADT, ExpressionTree ADT, Stack ADT, ExpressionTest.java. Compress these files into a single zip file and submit it with the following name:

```
zip proj4 ExpressionTest.java ExpressionTreeADT* BinaryTreeADT* StackADT*
cp proj4.zip /user/tvnguyen7/cs2400-003/bronconame-proj4.zip
```

NO package. You should check out your project on the CPP intranet using:

```
javac ExpressionTest.java
java ExpressionTest 1 2 + 3 /
```

Grading Guide:

- 10%: Project report and project output.
- 80%: Program correctness
- 10%: Coding – efficiency, style, comments, formats

Notes:

The following information is required in the beginning of every source file.

```
//
//    Name:        Last, First
//    Project:     #
//    Due:         date
//    Course:      cs-2400-03-sp21
//
//    Description:
//                 A brief description of the project.
//
```