CALIFORNIA STATE POLYTECHNIC UNIVERSITY
Computer Science Department
CS 241 02 (4) ***T. Nguyen/F17***

PROJECT: **1**
DUE DATE: **October 20, 2017**

Description:

Implement Chapter 24 Projects #3. Your *ExpressionTree* (ET) works with an expression that contains only variables or double values and supports operations +, -, *, and /.

Your ET must provide the following additional methods:
1. void setVariable(String name, double value) – set the variable *name* with a *value*
2. void displayPostfix() – output the ET in postfix format in one line, each token is separated by *one* space.

All exceptions must be handled properly.

Sample usage:

```
ExpressionTree expr = new
            ExpressionTree(new String[]{"a", "b", "2", "/","+"});
expr.setVariable("a", 1.5);
expr.setVariable("b", 2);
System.out.println(expr.evaluate());
expr.displayPostfix();
```

Required I/O:

```
F. Last's Expression Tree

…your program I/O
```

F. Last is your First initial and Last name.

Turn in:

1. Print out *only* the source code for ExpressionTree.java in landscape mode using non-proportional font (eg. Courier).
2. Compress all the source codes (*.java) into a single zip file with the structure:

```
proj1.zip
   ExpressionTest.java
   TreePackage/
        BinaryNode.java
        …
```

and submit it with the following name: *flast*-proj1.zip, where *flast* is your first initial and last name, *lowercase*.

```
cp proj1.zip /user/tvnguyen7/cs24102/flast-proj1.zip
```

You should check out your project on the CPP intranet using:

```
javac ExpressionTest.java
java ExpressionTest
```

Grading Guide:

- 80%: Program correctness
- 20%: Coding – efficiency, style, comments, formats

Notes:

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

```
//
//    Name:       Last, First
//    Project:    #
//    Due:        date
//    Course:     cs-241-02-f17
//
//    Description:
//                A brief description of the project.
//
```

2. The submission **must** be legibly printed.