EECS 233 Homework 5

General requirements:

- Due at 11:00 PM on the posted due date.
- Include your name and network ID as a comment at the top of all of your programs.
- Create a typed document (.txt or .pdf) with answers to the questions.
- Upload your document and all .java files as a .zip file to Blackboard. Do not use other formats such as .rar.
- All work should be your own, as explained in the Academic Integrity policy from the syllabus.

Instructions: This assignment requires you to work with expression trees and binary search trees. It requires you to use the programs ExpressionTree.java and BinarySearchTree2.java that were discussed (and provided) in class.

1. Revise ExpressionTree.java as explained below. Demonstrate these revisions and include sample output in your report.

   a. Create "compute" methods that compute the numerical value of an expression subtree. Create two methods in a manner similar to "printPostorder". One method receives the beginning node of a subtree. The other method receives no arguments but calls the first method recursively on the overall root.

   b. In your main method, ask the user for a fully parenthesized infix expression and use it to build the corresponding expression tree. Then use the new "compute" method to compute the result from the tree. Below is an example of the output (user input in **bold**). Assume the expression is entered correctly.

```
Enter an expression: (((2*(3+4))-5)/3)
Result: 3.0
```

2. In your report, draw the state of a binary tree after each of the following (in order a – n):

a. Insert 9	e. Insert 6	i. Insert 8	m. Remove 3
b. Insert 3	f. Insert 7	j. Remove 8	n. Remove 9
c. Insert 2	g. Insert 5	k. Remove 5	
d. Insert 1	h. Insert 4	l. Remove 6	

3. Revise BinarySearchTree2.java to have "printPreorder" and "printPostorder" methods, similar to "printInorder", but with preorder and postorder traversals, respectively. Demonstrate these revisions and include sample output in your report.