CS 260
Programming Assignment 5

This lab is worth a total of 100 points; 10 points are the self-evaluation, 50 points for the basic lab, 30 points for the advanced lab, and 10 points for a solution to the thinking problem.

Base Lab Specification

Write a class ParseTree that transforms a postfix algebraic expression into a parse tree as described in Moodle. The argument for the constructor is the string containing the expression. Operands are single letters, and the allowed operators are +, -, *, and /.

You can base your code off the pseudo code in the binary parse tree document in Moodle.

This lab should follow the course coding requirements and be split into multiple files as per your chosen language. There is a driver provided for this lab. You should test each part of your code as it is developed.

The base methods required are:

- constructor (Python init) – the parameter is a postfix expression as a string, create a tree from it. If the string is empty, create an empty tree.
- destructor (C++ only) – delete all nodes in the tree. This should be recursive.
- preOrder() – return a string containing a prefix version of the expression. This must be a recursive method.
- inOrder() – return a string containing an infix version of the expression, remember to add parentheses. This must be a recursive method.
- postOrder() – return a string containing a postfix version of the expression. This must be a recursive method.
- display() – displays the nodes in a tree graphically. The solution code for this is provided in Moodle under Supplemental Resources.

Advanced Lab Specification

For the advanced lab, start with your base lab class and add the following functionality.

- parseInOrder – pass in an infix expression as a string. Delete any nodes in the existing tree. Convert the infix expression to postfix. Then use the postfix to create a new tree.

Thinking problem

Inorder expressions need parentheses to deal with operator precedence. Modify the inOrder method to reduce the parenthesis to the minimum necessary.

*

*   -

For example, the output for this tree should be A*B*(C-D). Only the C-D term needs parentheses.