Umar Kagzi

Question 1 Whitepaper

In Question 1 of the HW, we are asked to comment all the code in both the “stack LL” and postfix.txt files. I have included the code and commented on both the files, but I just wanted to give a quick summary on how both files work.

Stack LL File

The code in this file is essentially an implementation of a linked list stack which has many benefits, the greatest of which is that fact that it can shrink or grow as much as the user wishes it to. Nodes can be dynamically allocate which is the highlight for a stack based LL.

The file starts out with declaring a template class type of linkedStackType which basically holds all of the stack objects such as push, pop, top, constructors and destructors, etc. After the declarations, we see the implementations of these template functions in detail. I have commented the code in detail, but it is pretty self-explanatory.

Postfix.txt File

The postfix.txt file is a program made to solve postfix expressions, often used in the computing field. Although these questions seem daunting at first, they are very simple to understand; it is just like doing regular algebraic calculations but you use the operation (multiplication, addition, etc) after you have inserted both numbers. For example, 4+8*3 would be written in postfix notation as: 4 8 3 * +, where the operators are written after the numbers in the expression.

To solve this, the program first takes in the expression via a file stream function. After reading the expression into its “calculator”, or the evaluateExpression function, it prints out the calculated result of the postfix expression in the user’s file. The evaluateExpression function is the heart of the program and checks if the program is in the correct format to parse using the evaluateOpr function that decodes which operators the postfix expression uses. The correct values are calculated one step at a time by inserting each value into the stack, calculating it using its specified operators, and finally printing the final value out using the printResult function.