Project 8 - User Document

When evaluating algebraic expressions a traditional order of operations is used to correctly evaluate the expression. However, this only works when the information is presented in a very specific order and the digits and symbols are in the correct positions. This method of presenting arithmetic expressions is referred to as infix notation. In computing though, there are various different notations that an arithmetic expression can be presented in. In this project expressions in a file are listed in prefix notation. Prefix notation is where the operators are presented in a manner in which they precede the operands. This notation is best evaluating using a stacked based algorithm that contains one stack that holds both operands and operators. For this program, the user is asked to input a data file that consists of the expressions that the user wishes to be evaluated. The file contains one expression per line and the end of the line signals the end of the expression. After the file is opened and read, the algorithm evaluates the expressions and displays the values to the terminal.

The program is split into 3 files. The main program is main.cpp, the header file is stack.h and the class file is stack.cpp. These files are located in the project8 folder underneath the programs folder.

To compile and link the files, enter:

g++ stack.h main.cpp stack.cpp

To run the program, enter a.out and respond to the program’s prompts for user input. This program will terminate after the list from the file are outputted to the terminal or a file that is not available is attempted to be opened.

After compiling the program and entering a.out an example run of the program would look something like this:

Enter input file name: postfix.dat

Expression: \* + 3 4 3

Value = 21.

Expression: \* 5 / 4 + 3 - 2 1

Value = 5.

EXPRESSION: \* 9 \* 2 - 5 + 2 4

Value = -18.

Note that the program terminates after all of the expressions have been evaluated.