**Programming Project Report**

Name: Blake Fasse

Date: July 20, 2017

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

The goal of this assignment was to get a better understanding of how to use the vector and stack classes by developing a rudimentary expression calculator. The program has the user input a mathematical expression such as (3 + 5) / 2 and covert that expression into a postfix expression 3 5 + 2 /. The program will then output the answer to that mathematical expression.

**Design:**

This assignment uses both the stack and vector classes that were designed in class. There were a total of three functions used: a function that implements the shunting yard algorithm, a function that evaluates the shunting yard algorithm, and a function that sets precedence to the operators in a mathematical expression. A pro to this set up is that it condenses the program by splitting the work of converting the input into a shunting yard equation, evaluating that shunting yard equation, and setting precedence to the operators between three different functions. A con of this set up is not using the built in stack and vector classes.

**Implementation:**

The only sample code that was given were prototypes for the functions, part of the main program, and the stack and vector classes that were used in class, if the classes count as sample code. The project description said to implement the shunting yard and evaluation functions. The shunting yard function will take the mathematical expression and push the numbers to the postfix vector immediately and add the operators to a stack. If there are any parenthesis then those will be discarded since postfix expressions do not require them but that does not mean they are worthless. If there are parenthesis around an equation then the first parenthesis will be put in the stack and the shunting yard function will go on as normal until the closing parenthesis where once it is found it will pop it then what’s on the stack until the opening parenthesis. The evaluation function try’s to solve the mathematical expression in the shunting yard algorithm form.

**Testing:**

Testing was done in two phases. The first phase was to make sure that the program did the math of a simple expression correctly. Next was to make sure that the program knew what to do if the functions had parenthesis. The normal inputs for these two phases were 2 + 2 and ( 2 + 2 ), respectively. Special cases that were used were ( 3 + 5 ) / 2, 3 + 5 ), and ( 3 + 5 with the last two cases being used to test the error checking. Everything worked as expected except for the case of ( 3 + 5 where the leading parenthesis is shown in the postfix expression.

**Conclusions:**

Except for the case where there isn’t a closing parenthesis around an expression, overall the program works exactly as expected. Even for that one case the math is done correctly it just shows the parenthesis in the postfix expression. I would say this project was a success. Next time I would use the built in stack and vector libraries instead of the ones used in class so I would get a better understanding of how to use them later. Development time took the whole two weeks.