**Design Documentation**

Introduction

The Infix calculator takes a infix equation ex. ( 7+(2\*8) ) and converts it to a postfix form via a stack which then is solved also using a stack. The program includes error checking to make sure only valid infix equations can be input.

UML

|  |
| --- |
| InfixExpression |
| - infix : string  - wellFormed(infix : string ) : bool  - balBrackets (infix : string ) : bool  - infixToPostfix () : string  - precedence ( op : char) : int  - postFixEval( postfix : String) : float |
| + InfixExpression()  + setValidInfix( infix : string ) : bool  + eval( ) : float |

Functions and its pseudocodes

The main program uses 2 main functions setValidInfix and eval these functions have access to the private member functions and do specific validation checks to make sure the infix equation has proper formatting. There are 5 private functions which check formatting, sets precedence for operators, converts infix to postfix and evaluates postfix equations.

Structure chart of the main program

Main program

InfixExpression

StackInterface

Node Class

Linkedstack

PrecondViolatedExcep

**Test Data**

|  |  |  |
| --- | --- | --- |
| **Explanation** | **Valid Inputs Values** |  |
| **Equation outside of the parentheses** | 3+5\*(2-8) | Solves the equation with respect to the parentheses. |
|  | Expected Output: -27 | -27 |
| **different operators in the parentheses** | 4-(4/2\*6) | Solves the equation with respect to the parentheses. |
|  | Expected Output: -8 |  |
| **Same operators in the parentheses** | 5\*(4/2/2)+9 | Solves the equation with respect to the parentheses. |
|  | Expected Output: 14 | 14 |
|  |  |  |
|  | **Invalid Inputs Values** |  |
| **Brackets are not balanced** | Input: -1+5 | There is no leading number so an error occurs. |
|  | Output: invalid input (prompts to try again) | Invalid msg displayed |
| **Equation is not well formed** | Input: 7\*(6+3 | The parentheses are not balanced. |
|  | Output: invalid input (prompts to try again) | Invalid msg displayed |

**User Documentation**

Description of the problem to solve

The program is designed to solve postfix equations with checks for formatting and allows the user to solve multiple equations.

File location on centOS

home/STCLOUDSTATE/it1032ao/CSCI301/Project5/task2

How to compile the program

Compiling the program is as simple as being in the file location on centOS and entering the following command “g++ main.cpp”.

How to run the program

After compiling the program, you are ready to run it. Type the command “./a.out”. This will prompt the user to enter an infix equation. If the equation entered does not have proper formatting it will ask the user to enter the equation again with proper formatting.

For Example:

If the user inputs: 7\*(6+3 the program will ask them to reenter the equation. If they were to then enter 7\*(6+3) the equation would be solved.