Assignment 3 – Part 2 & 3 Report

Part 2 and 3 due March 19 at midnight

Resources: parser.pdf, Bailey Bauman, Samantha Ray, Christopher Ridley, Abhishek Joshi,Chase Hinesman, William Flores, Ellie Miller,Caleb Johnson

1. Provide the design of your program for the parts 2 and 3 of the assignment. Write about the relation between classes and justification why you chose them.

   (a) In part 2 of the assignment, we were tasked with making templated versions of the LinkedStack and LinkedQueue classes. Both of these classes utilized the Templated DoublyLinkedList class to perform the operations of adding and removing elements through functions relative to each class. In part 3 of the assignment, we were tasked with constructing a class Parser which converts an expression from infix form to postfix form. We also constructed a class Evavluator which takes an expression in postfix form and performs the necessary calculations. Both Evaluator and Parser utilize the LinkedStack and LinkedQueue classes to perform their functions.

2. Describe each class private and public members, your algorithms and their implementations.

   (a) LinkedQueue
      i. DoublyLinkedList<T> ll: A DoublyLinkedList to be utilized by LinkedQueue
      ii. LinkedQueue<T>(){}: Constructor
      iii. ~LinkedQueue() { } :destructor
      iv. isEmpty(): Returns weather or not the LinkedQueue is empty
      v. first(): returns first element in the LinkedQueue
      vi. enqueue(): inserts an element at the end on the queue
      vii. dequeue(): removes an element at the fron of the queue

   (b) LinkedStack
      i. DoublyLinkedList<T> ll: A DoublyLinkedList to be utilized by LinkedQueue
      ii. LinkedStack<T>() {}: default constructor
      iii. isEmpty(): Returns weather or not the stack is empty
      iv. push (): inserts an element at the end on the stack
      v. top(): view top element
      vi. pop(): removes top element

   (c) Evaluator:
      i. *parser : pointer to the current stack
      ii. valStack: value stack
      iii. int eval(): two arguement evaluator
      iv. int eval(): one arguement evaluator
      v. isOperator: determines weather or not the token is an operator
      vi. Evaluator(): constructor
      vii. getValue(): gets value of the expression

   (d) Parser
      i. EnumTokens: Tokens to be used
      ii. static const char* delimiters: Delimeters to be used
      iii. LinkedStack<int> opStack: operator stack
      iv. LinkedQueue<std::string> postfix : postfix queue

v. char curVal: current value
vi. const char *tokens: arraay of tokens
vii. precTable[]: precedence table
viii. opTable[]: conversion table
ix. Parser(string s) : constructor
x. getPostfix(): returns postfix expression
xi. printPostfix(): prints postfix expression
xii. toPostfix(): converts infix expression to postfix expression

3. Write the names of the templated classes of your assignment. Which types have you used to test them for correctness?

   (a) LinkedStack: int, double, string, char

   (b) LinkedQueue: int, double, string, char

   (c) TemplatedDoublyLinkedList: int, double, string, char

4. Describe all tests done to verify correctness of your program for the parts 2 and 3 of the assignment. Give an explanation why you chose such tests. Include your tests in your assignment report.

   (a) When testing my program, I tested cases that would reflect the ability of my program to perform correct calculations involving addition, subtraction, multiplication, division, and multiple sets of parenthes both involving and not involvong the use of variables. I also made sure to reflect the ability of my program to detect unbalanced parenthesis and terminate when this error was caught.

   i.
```
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:06:35 03/21/17)
[:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o

[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:06:41 03/21/17)
[:: ./run-main
Does your infix expression include variables? (Y/N)
N
[1) Enter an infix expression: (6/3)*2+2
2) check if parentheis are balanced:
3) Display Infix expression: (6/3)*2+2
4) Display Postfix expression: 63/2*2+
5) Evaluate Postfix expression: 6/3=2
2*2=4
4+2=6
```

   ii.
```
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:05:08 03/21/17)
[:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o

[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:05:15 03/21/17)
[:: ./run-main
Does your infix expression include variables? (Y/N)
Y
enter a value for variable 'a':
4
[Enter an infix expression: (a+4)*3-1
2) check if parentheis are balanced:
3) Display Infix expression: (4+4)*3-1
4) Display Postfix expression: 44+3*1-
5) Evaluate Postfix expression: 4+4=8
8*3=24
24-1=23
```

iii.

```
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:16:46 03/21/17)
[:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o

[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:16:50 03/21/17)
[:: ./run-main
Does your infix expression include variables? (Y/N)
Y
enter a value for variable 'a':
7
[Enter an infix expression: (a-3+5)*2
2) check if parentheis are balanced:
3) Display Infix expression: (7-3+5)*2
4) Display Postfix expression: 73-5+2*
5) Evaluate Postfix expression: 7-3=4
4+5=9
9*2=18
```

iv.

```
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:11:17 03/21/17)
[:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o

[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:11:24 03/21/17)
[:: ./run-main
Does your infix expression include variables? (Y/N)
Y
enter a value for variable 'a':
5
[Enter an infix expression: (a*6)-4
2) check if parentheis are balanced:
3) Display Infix expression: (5*6)-4
4) Display Postfix expression: 56*4-
5) Evaluate Postfix expression: 5*6=30
30-4=26
```

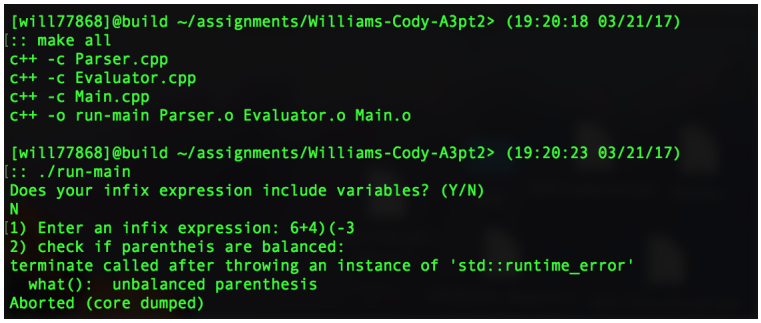
v.

```
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:13:57 03/21/17)
[:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o

[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:14:03 03/21/17)
[:: ./run-main
Does your infix expression include variables? (Y/N)
Y
enter a value for variable 'a':
8
[Enter an infix expression: 5*2-(a/4)
2) check if parentheis are balanced:
3) Display Infix expression: 5*2-(8/4)
4) Display Postfix expression: 52*84/-
5) Evaluate Postfix expression: 5*2=10
8/4=2
10-2=8
```

vi.

```
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:18:44 03/21/17)
:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o

[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:18:48 03/21/17)
:: ./run-main
Does your infix expression include variables? (Y/N)
N
1) Enter an infix expression: (4*5)-3*(4/2)
2) check if parentheis are balanced:
3) Display Infix expression: (4*5)-3*(4/2)
4) Display Postfix expression: 45*342/*-
5) Evaluate Postfix expression: 4*5=20
4/2=2
3*2=6
20-6=14
```

vii.


[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:20:18 03/21/17)
:: make all
c++ -c Parser.cpp
c++ -c Evaluator.cpp
c++ -c Main.cpp
c++ -o run-main Parser.o Evaluator.o Main.o
[will77868]@build ~/assignments/Williams-Cody-A3pt2> (19:20:23 03/21/17)
:: ./run-main
Does your infix expression include variables? (Y/N)
N
1) Enter an infix expression: 6+4)(-3
2) check if parentheis are balanced:
terminate called after throwing an instance of 'std::runtime_error'
what(): unbalanced parenthesis
Aborted (core dumped)