**CSCE 221 Assignment 3 Cover Page**

First Name Leilani Last Name Horlander-Cruz UIN 523008771

User Name leilanihc112 E-mail address leilanihc112@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more on Aggie Honor System Office website: http://aggiehonor.tamu.edu/

| Type of sources | | | | |
|---|---|---|---|---|
| | | | | |
| People | Ian Dickerson | | | |
| | | | | |
| Web pages (provide URL) | http://piazza.com | http://cplusplus.com | http://github.com | |
| | | | | |
| Printed material | | | | |
| | | | | |
| Other Sources | Powerpoint Slides | | | |
| | | | | |

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

*On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work*.

| Your Name | Leilani | | Horlander-Cruz | Date | 3-21-17 |
|---|---|---|---|---|---|

# CSCE 221 Assignment 3 – Parts 2 & 3

*Parts 2 & 3 due to CSNet by March 21*

**1 & 2. Design of program & Algorithms and Implementations**

LinkedList

This class consists of two pointers to ListNode: head and tail. Each ListNode contains two other members: obj, which is a string object used to store data; and next, a pointer to another ListNode. head serves as the first element in the linked list, while tail serves as the last element in the linked list. This forms a chain of elements, each node having a link to the next node.

LinkedStack

This class uses a linked list and forces LIFO behavior by limiting operations to pop, which removes the last element and returns it; top, which returns the first element; and push, which places an element onto the stack.

LinkedQueue
This class uses a linked list and forces FIFO behavior by limiting operations to dequeue, which removes the first element and returns it; enqueue, which places an element at the end of the list; and first, which returns the first element.

**3. Templated classes of assignment**

Given the difficulty I had during the project, my final implementation only supports integers. Integer numbers can be inputed in the form of "a+b/c*d" where a,b,c,d are all integer values. Letters can be used in place of integer values as well. Floating point numbers, however, do not work.

**4. Tests**

Given how my code only cooperates for integer values or letters representing integer values, I ran tests with numbers, letters, and operators. However, evaluation only worked for integer numbers. Infix to postfix expression worked for letters as well as integers. However, the precedence did not work fully in the evaluation process. Instead of evaluating ^ then * and / then + and -, it evaluated the expression completely left to right most of the time, ignoring precedence of the operators. When there was a power, it did the operation after it before doing the power operation.

The first one I ran was 1+3+2 which returned 13+2+ and evaluated as 6.
Second, I ran 1+2+8*9+3 which returned 12+89*+3+ which evaluated as 102.
Third, I ran 5^2-1 which returned 521-^ which evaluated as 5.
Similar tests to this were run. I could only test integers because the other types failed to work.