

**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				
Web pages (provide URL)	<a href="http://piazza.com">http://piazza.com</a>	<a href="http://cplusplus.com">http://cplusplus.com</a>		
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	3-1-17
-----------	---------	--	----------------	------	--------

## CSCE 221 Assignment 3 – Part 1

*Part 1 due to CSNet by March 1 with demonstration in labs on February 27/28.*

### Objective

This is an individual assignment which has three parts.

1. Part 1: C++ implementation of a Doubly Linked List for int type and next writing its templated version. The supplementary code is provided (download it from the class website).
2. Part 2: C++ implementation of queue and stack classes based on a templated Doubly

Linked List (implemented in Part 1).

3. Part 3: C++ implementation of a simple calculator for evaluating an algebraic expression based on its postfix form. The queue and stack classes (implemented in Part 2) should be applied for obtaining the postfix form and expression evaluation.

## Part 1: Implementation of Doubly Linked List

1. copy constructor -  $1+1+2+2+1+4n = O(n)$
2. assignment operator -  $2+2+1+4n+1 = O(n)$
3. output operator -  $1+2+1+5n+1+1 = O(n)$
4. insertFirst -  $2+1+1 = O(1)$
5. removeFirst -  $1+1+1+1+1+1+1 = O(1)$
6. first -  $1+1+1 = O(1)$

Testing DoublyLinkedList Main.cpp:

```
leilanihc112@build ~/221-A3-17a-code/DoublyLinkedList> (02:51:26 02/27/17)
: make
c++ -std=c++11 -c DoublyLinkedList.cpp
c++ -std=c++11 DoublyLinkedList.o Main.o -o run-dll

leilanihc112@build ~/221-A3-17a-code/DoublyLinkedList> (02:51:26 02/27/17)
: ./run-dll
Create a new list
list: {}

Insert 10 nodes at tail with value 10,20,30,...,100
list: {10 20 30 40 50 60 70 80 90 100}

Insert 10 nodes at front with value 10,20,30,...,100
list: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100}

Copy to a new list
list2: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100}

Assign to another new list
list3: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100}

Delete the last 5 nodes
list: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50}

Delete the first 5 nodes
list: {50 40 30 20 10 10 20 30 40 50}

Make sure the other two lists are not affected.
list2: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100}
list3: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100}

leilanihc112@build ~/221-A3-17a-code/DoublyLinkedList> (02:51:27 02/27/17)
: 
```

### Testing TemplateDoublyLinkedList TemplateMain.cpp:

```
[leilanihc112]@build ~/221-A3-17a-code/TemplateDoublyLinkedList> (02:32:55 02/27/17)
:: make
c++ -std=c++11 TemplateMain.o -o run-tdll

[leilanihc112]@build ~/221-A3-17a-code/TemplateDoublyLinkedList> (02:32:57 02/27/17)
:: ./run-tdll
Create a new list
list: {}

Insert 10 nodes at back with value 10,20,30,...,100
list: {10 20 30 40 50 60 70 80 90 100 }

Insert 10 nodes at front with value 10,20,30,...,100
list: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100 }

Copy to a new list
list2: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100 }

Assign to another new list
list3: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100 }

Delete the last 10 nodes
list: {100 90 80 70 60 50 40 30 20 10 }

Delete the first 10 nodes
list: {}

Make sure the other two lists are not affected.
list2: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100 }
list3: {100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100 }

[leilanihc112]@build ~/221-A3-17a-code/TemplateDoublyLinkedList> (02:32:59 02/27/17)
```

- **What to submit to CSNet?**
  - Your source code for DoublyLinkedList and TemplateDoublyLinkedList
  - Written report with complexity using big-O notation for all the functions.
  - The screenshots of testing all the cases should be included in your reports.