CSCE 221 Assignment 3 Cover Page

First Name Joseph Last Name Martinsen UIN 323009961

User Name josephmart E-mail address josephmart@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	Linked List Knowledge			
People				
Web pages (provide URL)				
Printed material				
Other Sources	CSCE 121 and 221 Lecture			

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 Joseph Martinsen Date 02/24/17

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.

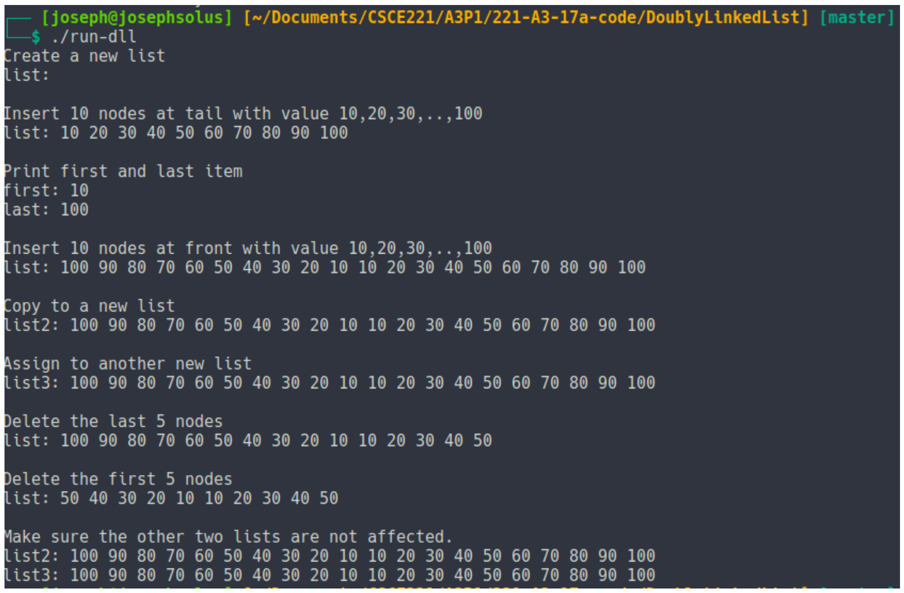
Part 1: Implementation of Doubly Linked List

- **Complexity Analysis**

  Comment each class member function you implemented with its time complexity using big-O notation. Specifically, comment on the loops.

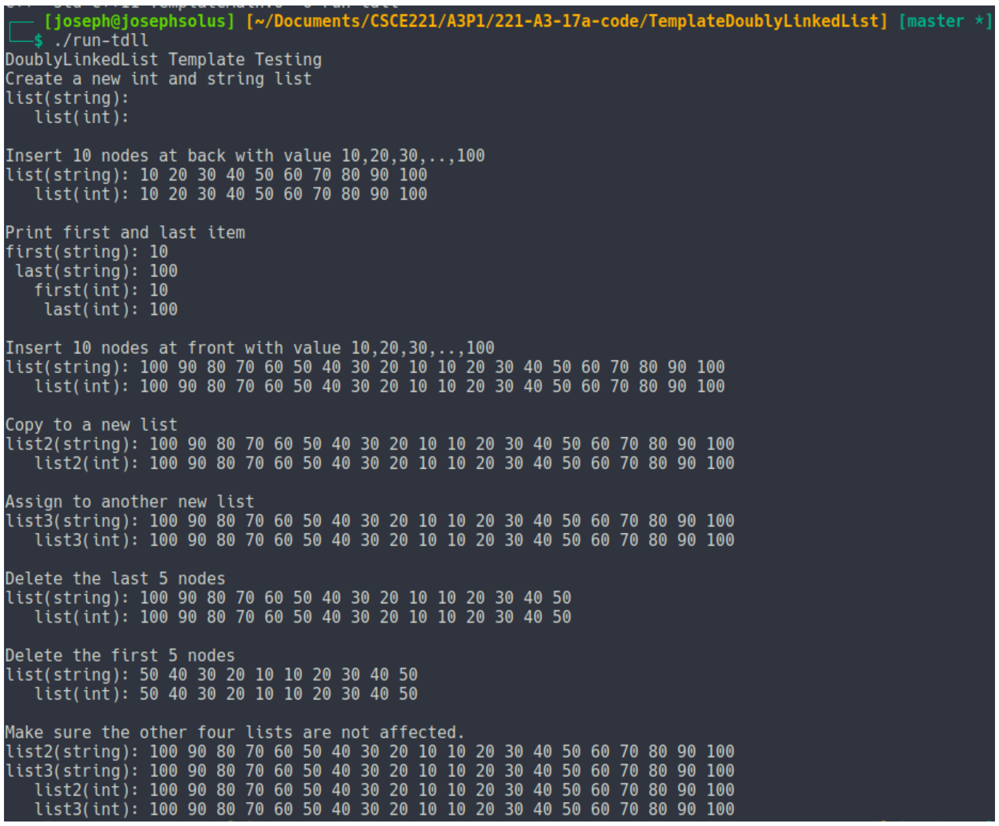
Table 1: Complexity Analysis

Function	**Big-Oh**
Copy Constructor	O(n)
Assignment Operator	O(n)
insertFirst()	O(1)
insertLast()	O(1)
removeFirst()	O(1)
removeLast()	O(1)
Destructor	O(n)
first()	O(1)
last()	O(1)
Output Operator	O(n)



```
┌── [joseph@josephsolus] [~/Documents/CSCE221/A3P1/221-A3-17a-code/DoublyLinkedList] [master]
└──$ ./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

Print first and last item
first: 10
last: 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
```

Figure 1: Doubly Linked Lists Tests



```
┌── [joseph@josephsolus] [~/Documents/CSCE221/A3P1/221-A3-17a-code/TemplateDoublyLinkedList] [master *]
└──$ ./run-tdll
DoublyLinkedList Template Testing
Create a new int and string list
list(string):
   list(int):

Insert 10 nodes at back with value 10,20,30,..,100
list(string): 10 20 30 40 50 60 70 80 90 100
   list(int): 10 20 30 40 50 60 70 80 90 100

Print first and last item
first(string): 10
 last(string): 100
   first(int): 10
    last(int): 100

Insert 10 nodes at front with value 10,20,30,..,100
list(string): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
   list(int): 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(string): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
   list2(int): 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(string): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
   list3(int): 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(string): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50
   list(int): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50

Delete the first 5 nodes
list(string): 50 40 30 20 10 10 20 30 40 50
   list(int): 50 40 30 20 10 10 20 30 40 50

Make sure the other four lists are not affected.
list2(string): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
list3(string): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
   list2(int): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
   list3(int): 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
```

Figure 2: Templated Doubly Linked Lists Tests