

### CSCE 221 Assignment 3 Cover Page

First Name    Suqian    Last Name    Wang    UIN    825009505

User Name    wangsuoqian123    E-mail address    wangsuoqian123@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	lecture sides			
People	Teresa Leyk			
Web pages (provide URL)				
Printed material				
Other Sources				

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    Suqian    Wang    Date    July 29th, 2017

# CSCE 221 Assignment 3

## Summer 2017

### 1. Linked list implementation

Doubly linked list:

Each node is connected to a front node and a back node using two pointers previous and next. Nodes were implemented using a class called DListNode. This class has a constructor, that initializes the value stored in each node, the previous node address, and the next node address.

The doubly linked list class is a lists that connecting all the DListNodes. This class's implementation included a constructor, a copy constructor, a destructor, assignment operator, functions to return the pointer to the first nodes and trailer, inserting and removing, overloading output operator, and return the length of the doubly linked list.

### 2. Complexity analysis

copy constructor:  $O(n)$  - initialized an empty list and go through the list while copying each element to another list

assignment operator:  $O(n)$  - clear existing linked list  $O(n)$  + copy the whole list from the other list  $O(n) = O(n)$

destructor  $O(n)$  - delete the whole list

insertFirst  $O(1)$  - create a node 1 + change the pointer 2 = 3 =  $O(n)$

insertLast  $O(1)$  - create a node 1 + change the pointer 2 = 3 =  $O(n)$

removeFirst  $O(1)$

removeLast  $O(1)$

first  $O(1)$  - return the first element

last  $O(1)$  - return the last element

insertAfter  $O(1)$  - create a node 1 + change the pointer 4 = 5 =  $O(1)$

insertBefore  $O(1)$  - create a node 1 + change the pointer 4 = 5 =  $O(1)$

removeAfter  $O(1)$

removeBefore  $O(1)$

DoublyLinkedListLength  $O(n)$  - traverse the whole list and increment the count.

overloading output operator  $O(n)$  - traverse the whole list and print out every node's content in the linklist

### 3. Testing results

```

SuqianWang — ssh -Y wangsuqian123@unix.cse.tamu.edu — 87x59
[wangsuqian123]@linux2 ~/Wang-Suqian-A3> (10:39:46 07/27/17)
[:: cd DoublyLinkedList

[wangsuqian123]@linux2 ~/Wang-Suqian-A3/DoublyLinkedList> (10:39:55 07/27/17)
[:: make
g++ -std=c++11 -c DoublyLinkedList.cpp
g++ -std=c++11 -c Main.cpp
g++ -std=c++11 DoublyLinkedList.o Main.o -o Main

[wangsuqian123]@linux2 ~/Wang-Suqian-A3/DoublyLinkedList> (10:40:00 07/27/17)
[:: ./Main
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

Insert 59 after the 1'th node in list2
list2: 100, 59, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90,
100

Insert 88 before the 1'th node in list2
list2: 88, 100, 59, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80,
90, 100

Remove the node after the 1'th node in list3
list3: 100, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Remove the node before the last node in list3
100, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 100

list:
The length for list is 0

list2: 88, 100, 59, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80,
90, 100
The length for list2 is 22

list3: 100, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 100
The length for list3 is 18

```

```
SuqianWang — ssh -Y wangsuqian123@unix.cse.tamu.edu — 87x59
DoublyLinkedList TemplateDoublyLinkedList

[wangsuqian123]@linux2 ~/Wang-Suqian-A3> (10:45:20 07/27/17)
[:: cd TemplateDoublyLinkedList ]

[wangsuqian123]@linux2 ~/Wang-Suqian-A3/TemplateDoublyLinkedList> (10:45:29 07/27/17)
[:: make ]
g++ -std=c++11 TemplateMain.cpp -o TemplateMain

[wangsuqian123]@linux2 ~/Wang-Suqian-A3/TemplateDoublyLinkedList> (10:45:32 07/27/17)
[:: ./TemplateMain ]

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
Insert 59 after the 1'th node in list2
list2: 100, 59, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90,
100

Insert 88 before the 1'th node in list2
list2: 88, 100, 59, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80,
90, 100

Remove the node after the 1'th node in list3
list3: 100, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Remove the node before the last node in list3
100, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 100

list:
The length for list is 0

list2: 88, 100, 59, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80,
90, 100
The length for list2 is 22

list3: 100, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 100
The length for list3 is 18
```