Programming Assignment 3 Part I
Carson Hanel

Preface: Alright, I realize this isn't LyX but I thought this would be nicer to look at than a shoddily put together LyX generated PDF. Not much in the way of guidelines was provided, so I'm sort of winging this, but trying to hit the major points. I'll probably copy and paste this to the head of the other page as well. It is highly helpful to pair this document with my code comments.

1. Assignment description:
    a. The assignment was to implement the STL Doubly Linked List, both specifically for integers and as a templated class. The Doubly Linked List is an organization of data utilizing data members header and trailer which are data members called "Dummy Nodes" which hold more immediate access to the two ends of the list. A Doubly Linked List is different, and separate from, a Singly Linked List in that it is bidirectional, and in this case, utilizing dummy variables. Doubly Linked Lists can be characterized by the utilization of pointers connecting data members. This is unique from an array in that disassociation, reassociation, and ordering can be done in-place, however this is disadvantageous. The Doubly Linked List suffers of inefficiency because it can only be accessed in a linear fashion unless accompanied by an array of pointers.
2. Description of DS&A used:
    a. The data structures used include:
        i. The class Node
        ii. The class DoublyLinkedList
    b. The algorithms used include:
        i. Linear search using pointers
        ii. Constant removal from beginning or end
        iii. Linear removal from internal nodes
    c. How I solved the problems:
        i. Refer to commenting on code. Missing comment sections are because I was having a bit of trouble working out some pointer issues earlier, and had to reimplement a few sections.
3. A user guide to navigate my program:
    a. Access the DoublyLinkedList folder
    b. Type make into the environment
    c. Type ./a.out
4. Types of exceptions:
    a. Trying to delete from areas that cannot be deleted from:
        i. Empty list
        ii. Deleting after, when the node is the last node
        iii. Deleting before, when the node is the first node
5. Testing:
    a. Non templated:

i.

compute.cse.tamu.edu - PuTTY

```
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,

Inserting 2048 after node 100
Assigning pointers
Temp's pointers assigned
Happened after me
Pointers to temp assigned
Inserted.
list: 100, 2048, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70,
 80, 90, 100,

Inserting 1024 before node 100
list: 100, 2048, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70,
 80, 90, 1024, 100,

Removing after the first
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 9
0, 1024, 100,

Removing before last
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 9
0, 100,

Object being checked for length: 0x742290 with value: 100
Object being checked for length: 0x7422b0 with value: 90
Object being checked for length: 0x7422d0 with value: 80
Object being checked for length: 0x7422f0 with value: 70
Object being checked for length: 0x742310 with value: 60
Object being checked for length: 0x742330 with value: 50
Object being checked for length: 0x742350 with value: 40
Object being checked for length: 0x742370 with value: 30
Object being checked for length: 0x742390 with value: 20
Object being checked for length: 0x7423b0 with value: 10
Object being checked for length: 0x7423d0 with value: 10
Object being checked for length: 0x7423f0 with value: 20
Object being checked for length: 0x742410 with value: 30
Object being checked for length: 0x742430 with value: 40
Object being checked for length: 0x742450 with value: 50
Object being checked for length: 0x742470 with value: 60
Object being checked for length: 0x742490 with value: 70
Object being checked for length: 0x7424b0 with value: 80
Object being checked for length: 0x7424d0 with value: 90
Object being checked for length: 0x7424f0 with value: 100
The length of the list2 is: 20

[chingyl510]@compute ~/221/P3/DoublyLinkedList> (23:07:47 07/29/17)
::
```

compute.cse.tamu.edu - PuTTY

```
[chingy1510]@compute ~/221/P3/DoublyLinkedList> (23:07:43 07/29/17)
:: ./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, 9
0, 100,


Deleted DLL
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,

Inserting 2048 after node 100
Assigning pointers
Temp's pointers assigned
Happened after me
Pointers to temp assigned
Inserted.
list: 100, 2048, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70,
 80, 90, 100,

Inserting 1024 before node 100
list: 100, 2048, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70,
 80, 90, 1024, 100,

Removing after the first
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 9
0, 1024, 100,

Removing before last
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 9
0, 100,
```

b. Templated:

i.

```
[chingyl510]@compute ~/221/P3/TemplateDoublyLinkedList> (23:19:02 07/29/17)
:: ./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,

Inserting ABC after node 100
Inserted.
list: 100, ABC, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Inserting XYZ before node 100
list: 100, ABC, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, XYZ,

Removing after the first
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, XYZ, 100,

Removing before last
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

The length of the list2 is: 20
```