

CS311 Yoshii HW3 Part 1 - Linked List Class (based on weeks 5-6)

DUE: Week 7 Friday

TOTAL: 30 points Your score is:

Your NAME: Alec Guilin

Date turned in: October 13 2017

Purpose: Review and improve your linked list.

PRE-PROGRAMMING TASKS [12pts]

Your score is:

These 11 points are from in-class pointer and list exercises on Week 5 Thursday (5pts) and Week 6 Tuesday (7pts) - they had to be submitted on these dates.

PROGRAMMING: Linked List class [2+16=18pts]

Your score is:

This is a singularly linked Linked List class based entirely on the Week 6 notes.

Do not submit the linked list class you created for cs211. You will get 0 points.

All data members, and function names must match the cs311 notes from Week 6 or you will lose points.

*****Compile llist.h and llist.cpp by Monday!!!*****

llist.h, llist.cpp, HW3P1client.cpp templates were provided. Must use them!!

Comments must include the following or you will lose points:

Every special case should be commented.

e.g. // the case where this is the first node

Every local variable should be commented with its purpose.

e.g. // P will be used to point to the second to the last node

Client/Main: A menu based program

Exceptions should not abort the program.

Case 1:

1. check empty and report the result
2. display the list L.displayAll();
3. add 4 integers L.addRear(1); L.addRear(2); L.addRear(3); L.addRear(4)
4. display the list L.displayAll(); - 1 2 3 4

5. remove from front twice (and display the elements as they are removed)
6. display the list - 3 4
7. check empty again and report the result
8. remove from **the rear** twice (display the elements removed)
9. check empty again and report the result

Case 2:

1. add to front once (element 5)
2. add to front again (element 4)
3. delete Front -- this removes 4
4. add to rear 3 times (elements 6,8,9)
5. displayAll (4 elements) - 5 6 8 9
6. add before the 1st (element 4) - 4 5 6 8 9
7. add before the 4th (element 7) - 4 5 6 7 8 9
8. add before the 7th (element 10) - 4 5 6 7 8 9 10
9. add before the 9th (element 12) - **error (out of range)**
10. add before the 0th (element 0) - **error (out of range)**
11. displayAll - 4 5 6 7 8 9 10
12. delete Ith I==1 (indicate the element removed) - 5 6 7 8 9 10
13. delete Ith I==6 (indicate the element removed) - 5 6 7 8 9
14. delete Ith I==3 (indicate the element removed) - 5 6 8 9
15. delete Ith I==5 - **error (out of range)**
16. delete Ith I==0 - **error (out of range)**
17. displayAll - 5 6 8 9
18. delete from rear until it is empty (indicate the elements removed)
19. displayAll - [empty]

Case 3:

1. add before the 0th - **error (out of range)**
2. delete front - **error (underflow)**

Case 4:

1. delete 2nd - **error (out of range)**
2. delete rear - **error (underflow)**

Note: The above are the minimal required test cases. You should test other error cases thoroughly since this program will be used in later assignments!

Q: State of the Program:[2pts]

- Does your program compile without errors? If not, describe: Yes.
- List any bugs you are aware of, or state “No bugs”: The only bug I found is in the 13th test case in case function 2. It deletes the correct node but for some reason it doesn’t cout the correct value for the oldNum. This function works perfectly fine in other instances of the program when called and it has the correct value from OldNum so I’m not really sure what is causing this issue.

Submit these 5 files:

1. This assignment sheet with answers inserted.
2. Source code (l1ist.h, l1ist.cpp, HW3Plclient.cpp)

3. Script (Test) of compilation and testing of the 4 cases in the given order

Whether working or not, test result must include the lines for compiling your files or we will not grade your program i.e. 0 points for the program.