CS 311 - HW 5 - 100 points

**Lined list**

Write a **Linked List** application program to work with linked lists.

**Steps:**

1. Compile and run the files at HW5demos folder. Write down the results of your runs.
2. Implement the given llist class: complete the attached **llist.h** and **llist.cpp**
3. Refer to Lecture 11 and HW5\_help.docx for details of implementation.
4. Note that your grade will be 0 if you submit some other implementation of linked list class.
5. Note that you will need this implementation of linked list class for HW6 as well. Your HW6 will not be graded if your submission to HW5 does not work.
6. Note that all data members and function names must match **HW5\_help.docx**
7. Comment your implementation and include the following in your comments.
   * 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
8. Complete the attached menu-based program **HW5client.cpp**
9. Note that in this program, exceptions should **NOT** abort the program.
   * Exception handling is necessary.
10. Test your program using the following **Test1** and enter the results to your **test1.txt**. These test cases are also implemented in HW5client.cpp

**Test1:**

**Case 1 – Basic:**

1. check empty and report the result
2. display the list L.displayAll();
3. add 4 integers **to rear** 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 – Insertion and Deletion:**

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

**Case 3 – Overloading and Copy Constructor:**

1. Create a 5 element list with 1,2,3,4,5 (L1)
2. Pass the list to a **client function** called CopyTest to test your copy constructor.
   1. Copytest will receive the list **passed by value** from main() and
   2. Simply 1) add a node to its rear with 6 in it (should not affect the original)

2) display it (6 elements 1,2,3,4,5,6)

1. Display L1 (this should still be a 5 element list)
2. Do L1 = L1;
3. Display L1 (this should still be 1 2 3 4 5)
4. Create a 4 element list L2 with 7,8,9,10.
5. Display L2
6. Do L2 = L1; (L2 becomes 5 elements 1,2,3,4,5)
7. Display L2.
8. Remove a rear node from L1. (This should not affect L2).
9. Display L1. (L1 is 1,2,3,4)
10. Display L1 again. (4 elements – just to make sure)
11. Display L2 again. (still 5 elements 1,2,3,4,5)

**Submission**

Submit a zip file containing the following files.

1. llist.h (5 points) -- class header file
2. llist.cpp (60 points) -- class implementation file
3. HW5client.cpp (30 points) -- the implemented application
4. test1.txt (5 points) -- results of Test1

Important note1: You will miss up to 10 points if you don’t comment your programs.

Important Note2: Always make sure the files you submit can be compiled on **empress.csusm.edu** with no error. We will compile and test your files on empress.