**CS311 Yoshii HW3 Part 1 - Linked List Class (based on Notes-6A)**

**DUE: 3/23/2020**

**TOTAL: 20 points Your score is:**

**Fix and resumit with P2?**

**Your NAME:**

**Date turned in:**

**========================================================**

**PROGRAM: Linked List class [4+16=20pts] Your score is:**

**========================================================**

**Header:**

**Implementation:**

**Client:**

**Test results:**

**Total 16 points:**

**Q’s 4 points:**

**Q1: Observe and write down the results from equalstest.cpp, getstest.cpp, and copyconsttest.cpp [2pts]**

**Q2: State of the Program:[2pts]**

* **Does your program compile without errors? If not, describe:**
* **List any bugs you are aware of, or state “No bugs”:**
* **Make sure you have aleady used the** [**Pointer**](http://cgi.csusm.edu/ryoshii/MyVisualizers/pointstr.html) **and** [**Linked List Visualizers.**](http://cgi.csusm.edu/ryoshii/EnhancedLLVTutor/menu/visualizer/)
* **Complete my llist.h and llist.cpp.**

1. **Do not submit the linked list class you created for cs211. You will get 0 points.**
2. **All data members, and function names must match HW3P1\_help or you will lose many points.**
3. **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**

* **A menu based program HW3P1client.cpp was provided. Must complete it!**

**Exceptions should not abort the program.**

**Required Test Cases: 🡺 Test.txt**

**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) and displayAll – 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) ; displayAll - 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)

**This program will be used over and over again in this class so you MUST make it perfect!!!!!**

**I will not grade P2 if your P1 is not working.**

**SUBMIT THESE 5 FILES:**

1. **This assignment sheet with answers inserted.**
2. **Source code (llist.h, llist.cpp, HW3P1client.cpp)**
3. **(Test) of compilation and testing of the 3 cases in the given order.**

**CHECK LIST before you submit:**

* **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.**
* **Did you answer all the questions?**
* **Did you check your comments and style against CS311 How To Comment.doc??**