CS311 Yoshii HW3 Part 3 - Linked List Extended (based on Week 7)

==

DUE: Week 8 Friday

TOTAL: 20 pts **Your score is:**

***Your Name:** Alec Guilin

***Date Turned In:** 10/21/17

***Did you do the extra credit?[5pts] <answer here or we will not look for it>**

Purpose: To be able to implement and use an overloaded operator and a copy constructor to complete the linked list class.

Preparation Questions [2pts]:

Q1: What was the result of compiling overloadtest1.cpp?
error: invalid operands to binary expression ('llist' and 'llist')
Q2: What was the result of compiling overloadtest2.cpp?
Compiled and ran without any errors. However to see if A is the same as B we would need to display the two lists after the = operator to see what really happened.

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

==================================

Finally, add the following to your Searchable List (slist) class from HW3P2.
Pseudo code was already provided for you.

overload operator = (see Week7 Notes-7A notes)

create a copy constructor (see Week7 Notes-7A notes)

Client (HW3P3client.cpp):
Your client file will have two functions: Main() and CopyTest
Main will

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.
    a. Copytest will receive the list **passed by value** from main() and
    b. 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)
3. Display L1 (this should still be a 5 element list)
4. Do L1 = L1;
5. Display L1 (this should still be 1 2 3 4 5)

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

It is very important to complete this linked list class and make sure the = and copy constructor work perfectly. You will use this to implement HW6 and HW7 graphs.

Q3) State of the program [2pts]

- **Does your program compile without errors?** Yes.
- **List any bugs you are aware of, or state "No bugs":** No bugs

Submit these 5 files:

1. **this assignment sheet with your answers**
2. **slist.h - the updated header**
3. **slist.cpp – the updated implementation**
4. **HW3Pclient.cpp - the client file**
5. **Test - the script of compilation and test results (one run)**

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

EXTRA credit [5pts] – highly recommended to do
Must work perfectly to receive any points. No partial credit. So, if it does not work perfectly, do not submit it.

Overload operator == by adding it to the above slist.h and slist.cpp

Create a separate client file hw3ec.cpp to test ==.

Test cases – right before displaying the result of each comparison, must display the following description of the case:

1. L1 is empty and L2 is empty	**true**
2. L1 is empty and L2 has 2 elements	**false**
3. L1 has 2 elements and L2 is empty	**false**
4. L1 has 1,2,3 and L2 has 1,2,3	**true**
5. L1 has 1,2,3 and L2 has 1,2	**false**
6. L1 has 1,2,3 and L2 has 1,2,3,4	**false**
7. L1 has 1,2,3 and L2 has 1,2,4	**false**

Submit these 2 additional files:

1. **Client for extra credit – hw3ec.cpp (you should have the == in the original header and implementation you submitted above)**
2. **Test results for extra credit - TestEC**