**Linked List lab (only head pointer)**

1. Get dropbox code. Read through all of the starter code. The constructor and print() are already written for you.

2. Write prepend(). This is the easiest of the functions that add to the linked list --- this adds an element at the front (head) of the list. What is the big-oh time? Write some code in main() to test this function to make sure it works.

3. Write push_back(). This is harder, because you need to traverse the list to find the last node. Make sure you don't walk off the end of the list! Hint: there are two cases here, one where the list is empty, and one where it is not. What is the big-oh time? Write tests in main.

4. Write remove_data(). This searches for and removes an item in the linked list by its data value, as opposed to position. You will need to traverse the list.

5. Write the destructor. The destructor should traverse the list and delete all the nodes within. This is tricky! Don't delete a node if you plan on accessing its next pointer immediately after!

6. Write size(). A C++ vector has a function called size(), so our linked list should have the same functionality. We haven't explicitly talked about how to write this, but I'm sure you can figure it out. What is the big-oh time?

7. Write at(). This should work similarly to the C++ vector at() function. What is the big-oh time?

8. Write insert(). This function will allow insertion into the middle of a list.