

## CS260 - Programming Assignment 1 - Linked List Implementation and Use

The purpose of this assignment is to give you some practice with implementing and using a linked list in its most basic form.

The next assignment will build on this and have you turn your linked list code into a template so you can reuse the code for objects of any type.

---

### Overview

For this assignment you'll write a console application that keeps track of names (strings) on a list.

When complete, you'll have the following classes/structs (your class names may be different):

- LinkedList (example was done in class, and another is in the book)
- Node (example was done in class, and another is in the book)

You should have a .h and .cpp file for the LinkedList. And maybe Node, although Node could conceivably be a struct and/or need no member functions, so it could be embedded inside the LinkedList files (no programs using the LinkedList class would need to really know about the Node or see an interface to it).

And you'll have some code (main, possibly a few other functions) to implement the user interface. All menu/error output should occur in this code; the only output done in LinkedList or Node is streaming the items in the list to cout.

Important Notes:

- You shouldn't use a template as that will be the next assignment.
  - You may NOT use any templates from the Standard Template Library. We'll do that later in the term.
  - The user interface (including main()) should be in a separate source file. Maybe called stringList.
- 

### Details

The linked list object should support the following methods (public functions):

- insert - inserts the given item in the list if it's not already there. Returns true if successful.
- remove - removes the given item from the list. Returns true if successful.
- print - prints the list in-order
- count - returns a count of the number of items in the list.
- removeAll - removes all items from the list
- printReverse - prints the list in reverse order (optional)

Your linked list should store a list of strings in alphabetical order. Duplicates are not allowed.

You should display a menu to the user consisting of the following options:

- a - Add an item
- r - Remove an item
- d - Remove all items from the list (might want to ask for confirmation)

Additional details

- You should have a constructor and destructor (destructor should delete every node)
  - If an item is to be added or removed, prompt for the item name to be added or deleted (spaces should be allowed).
  - After each operation:
    - give the user feedback as to whether the operation was successful or not
    - re-display the list
    - redisplay the menu.
- 

## **Very Important Stuff**

Program should be well written, function properly, and be both easy and efficient to use.

All programs should follow the class's [Coding Conventions](#)

Submit the following:

- A zip file containing all your .cpp and .h files and your executable
- The executable as a separate file (so the .exe gets uploaded twice)