

CSE 1384 - Linked Lists

Lab 7

Objectives:

- Continue practicing past concepts
- Practice using linked lists in C++

Assignment:

For this assignment, we'll be practicing using linked lists. You'll begin the assignment with a starter file that you should complete (`linkedlist.cpp`) and a few files you **shouldn't change** (`linkedlist.h`, `node.h`, `main.cpp`, `books.txt`). The driver file (`main.cpp`) will serve as a testing file; your output should match mine exactly if working correctly.

The `LinkedList` class has some initial functions already defined:

- Constructor
- Destructor
- Display - displays the linked list
- Append - adds an item to the end of the linked list
- Pop - removes the last item from the linked list

You should not change these functions already defined. Your job will be to complete the `LinkedList` class to make the driver (`main.cpp`) fully functional. You can use these functions as a basis to begin creating your own functions. The linked list will be a list of books (containing two string variables: a title and an author).

A text file provided to you must be placed where the main can open it and create the initial list of books. You don't have to code the file interactions, this is done for you. Just make sure you put `books.txt` in the correct location.

The node header file (`node.h`) defines your node and your optional constructors available to you. The linked list header file (`linkedlist.h`) has definitions for the functions you should be adding to the code (in the `linkedlist.cpp`). The return values and parameters are listed appropriately there. These functions are:

- insert
 - Insert a node at the position sent. So, if position 1 is chosen, it'd be like index 1 in an array. Size should increase (AKA, the item should NOT overwrite the current)
- remove
 - Remove a node the position sent (starting at 0)
- view
 - View the book at the position sent (starting at 0)

Error Checking:

Take note that the insert, remove, and view functions have a return type of an integer. You should be returning 1 upon a failed operation and 0 upon a successful operation. Indexing starts at 0.

Hints:

Insert:

You should be able to handle the following: inserting to the head, inserting to the end of the list, inserting in the middle of the list. Consider keeping track of the item before the one you're inserting and the item in the current place you're inserting to.

Remove:

You should be able to handle the following: removing from the head, removing from the end of the list, and removing from the middle of the list. Consider keeping track of the item before the intended deleted item and the item after the deleted item. *Don't forget to explicitly delete the item using keyword "delete" to avoid memory leaks.*

Milestones:

You must receive a check-off for each item to receive full credit on the lab. You may show the milestones incrementally, or receive a check off for multiple in one demo depending on your progress.

Examples of each milestone are included. For this lab, the initial state of your output should be:

```
Initial list structure:
Book 0: The Color Purple by Alice Walker
Book 1: The Road by Cormac McCarthy
Book 2: Gilead by Marilynne Robinson
Book 3: To Kill a Mockingbird by Harper Lee
Book 4: 1984 by George Orwell
Book 5: Beloved by Toni Morrison
Book 6: Misery by Stephen King
```

Milestone 1:

Error checking. To the view, insert, and remove functions add the appropriate error checking.

Must show:

- All error messages working
- One doesn't break the list
- Your output matches mine

```
Error checking:
Not a valid remove -- out of range!
Not a valid remove -- out of range!
Not a valid insert -- out of range!
Not a valid insert -- out of range!
Not a valid view -- out of range!
Not a valid view -- out of range!
```

Milestone 2:

Insert capabilities added to the LinkedList. Must show:

- Head, tail, and middle inserts working
- Your output matches mine

```
Display after inserts:
Book 0: Brave New World by Aldous Huxley
Book 1: The Color Purple by Alice Walker
Book 2: The Road by Cormac McCarthy
Book 3: Gilead by Marilynne Robinson
Book 4: To Kill a Mockingbird by Harper Lee
Book 5: Animal Farm by George Orwell
Book 6: 1984 by George Orwell
Book 7: Beloved by Toni Morrison
Book 8: Misery by Stephen King
Book 9: I, Robot by Isaac Asimov
```

Milestone 3:

Remove capabilities added to the LinkedList. Must show:

- Head, tail, and middle removes working
- Your output matches mine

```
Display after removes:
Book 0: The Color Purple by Alice Walker
Book 1: The Road by Cormac McCarthy
Book 2: Gilead by Marilynne Robinson
Book 3: Animal Farm by George Orwell
Book 4: 1984 by George Orwell
Book 5: Beloved by Toni Morrison
```

Milestone 4:

View capabilities added. Must show:

- Output matching mine

```
Viewing a specific book:  
Animal Farm by George Orwell
```

Milestone 5:

All the functions work fine with mine! They didn't break each other or anything else. This can be found before now, but this can be the make it or break it. There's two things in the code that count for this one. Must show:

- Both items with the output matching mine
- ENTIRE program output correct (see entire program output on next page)

```
Display after a pop and append:  
Book 0: The Color Purple by Alice Walker  
Book 1: The Road by Cormac McCarthy  
Book 2: Gilead by Marilynne Robinson  
Book 3: Animal Farm by George Orwell  
Book 4: 1984 by George Orwell  
Book 5: Frankenstein by Mary Shelley
```

```
List was empty -- pop failed!
```

```
Display attempt after empty:  
List was empty -- display failed!
```

Comment Block:

Your code should contain a comment block at the top containing information on who wrote the code, what the assignment is, when it is due, etc. Here is an example of a good comment block to put:

```
/*  
Name: <your name>                                NetID: <your netID>  
Date: <current date>                            Due Date: <enter in due date>  
  
Description: <What is the program?>  
*/
```

Entire Program Output (for reference):

```
Initial list structure:
Book 0: The Color Purple by Alice Walker
Book 1: The Road by Cormac McCarthy
Book 2: Gilead by Marilynne Robinson
Book 3: To Kill a Mockingbird by Harper Lee
Book 4: 1984 by George Orwell
Book 5: Beloved by Toni Morrison
Book 6: Misery by Stephen King

Error checking:
Not a valid remove -- out of range!
Not a valid remove -- out of range!
Not a valid insert -- out of range!
Not a valid insert -- out of range!
Not a valid view -- out of range!
Not a valid view -- out of range!

Display after inserts:
Book 0: Brave New World by Aldous Huxley
Book 1: The Color Purple by Alice Walker
Book 2: The Road by Cormac McCarthy
Book 3: Gilead by Marilynne Robinson
Book 4: To Kill a Mockingbird by Harper Lee
Book 5: Animal Farm by George Orwell
Book 6: 1984 by George Orwell
Book 7: Beloved by Toni Morrison
Book 8: Misery by Stephen King
Book 9: I, Robot by Isaac Asimov

Display after removes:
Book 0: The Color Purple by Alice Walker
Book 1: The Road by Cormac McCarthy
Book 2: Gilead by Marilynne Robinson
Book 3: Animal Farm by George Orwell
Book 4: 1984 by George Orwell
Book 5: Beloved by Toni Morrison

Display after a pop and append:
Book 0: The Color Purple by Alice Walker
Book 1: The Road by Cormac McCarthy
Book 2: Gilead by Marilynne Robinson
Book 3: Animal Farm by George Orwell
Book 4: 1984 by George Orwell
Book 5: Frankenstein by Mary Shelley

Viewing a specific book:
Animal Farm by George Orwell

List was empty -- pop failed!

Display attempt after empty:
List was empty -- display failed!
```

Deliverables:

- C++ code (.cpp file)
 - If you completed everything, just submit "linkedlist.cpp"
 - If you didn't complete everything make sure you submit your working main.cpp as well

Point Breakdown:

(100 points total)

A submission that hasn't received a check-off will not be considered for grading.

Grading Note: indexing must start at zero. If it doesn't, it's not adding/removing in the correct locations and that is an incorrect add/remove.

- 10pts - milestone 1
- 20pts - milestone 2
- 20pts - milestone 3
- 10pts - milestone 4
- 30pts - milestone 5
- 10pts - programming style *
- (PENALTY) 15pts - changing the code given to you (main.cpp, node.h, linkedlist.h, OR any of the existing functions in the linkedlist.cpp)

* Programming style includes good commenting, variable nomenclature, good whitespace, etc.

(Commenting out non-working portions of the main.cpp is the only acceptable change)