Roles

Leader (Mal Flesher)

- Lead
- Manage
- Organize
- Driver.cpp & README.md
- Formal Write up

Architect (David Fields)

- Constructs files
    - Headers
    - .cpp
- Delegate Files task to each individual
- LinkList.cpp/.hpp

Debugger/Tester (Noah Carter)

- Debug Code
- Test code
- Delegate and organizes fixes in the code and delegates it as such
- county.cpp/.hpp & counties.txt

Program Description

Idea

- Imaginary County List

Classes

- Linked List Class
- Storage Class
- Other Class – list node (inside of linkedlist.hpp)

What did your group try to do?

- We originally were going to try a smart pointer, but after discussion on what that was and what we needed to do, we decided on a list node as we understood the code for that better.

What was successful?

- Meeting up and programming our different parts in the same room. It allowed us to ask questions and help each other out when we needed. It was a little harder to figure out times where everyone could meet outside of lab time, but it helped in the long run.
- Git made it very easy to pass code between each other.

What was unsuccessful?

- The original way we were trying to write our other file, now called county but was called storage first. We originally were trying to write it like a storage file which was making everything a lot harder than it needed to be.
- Our save and load function

What was your process?

- Someone made the basics of the header files and created the .cpp files we needed. Then we all took induvial role of a file to write before coming together to debug and fix what was needed.

What would you do differently?

- I would be a little better about communicating what part everyone is working on and the expected outcome of said part, and to be more vocal about code issues before they get too annoying for one person to look at. Me, I am the issue, I need to ask my questions before I cry.

What did you learn?

- I got a better understanding of templates and how to use/recall them.

What is the expected outcome of the program?

- The program should start with a menu with 8 different options and should not let you pick a number outside of that range. It should continuously run until you pick the 8th option that ends the program.

1. Add to end of list
2. Add to front of list
3. Add to anywhere in list
4. Remove from anywhere in list
5. Sort list
6. Save list (I'm sorry this no work)
7. Load previous list (I'm sorry this no work)
8. End Program

How will the TA run/test your program?

- Compile and run program to test each file. Use counties.txt to test the save and load functions. Run through each option at least once, except for the save and load functions, they no work