Project 3: Linked List

Start Assignment

* **Due** Mar 22 by 11:59pm
* **Points** 50
* **Submitting** a file upload
* **File Types** c
* **Available** Feb 24 at 12am - Mar 22 at 11:59pm 27 days

**Linked Lists**

A linked list is a sequence data structure consisting of a chain of nodes. Each node contains both data and a pointer to the next element in the list. These elements are not necessarily stored in contiguous memory locations

**Corrections Additions and Important Notes!**

* We’re using the Head struct variant on page 26 of the reading! See the provided template.

**Files**

* [**linked\_list.c**](https://canvas.wisc.edu/courses/280031/files/24893967?wrap=1)
* [**output.txt**](https://canvas.wisc.edu/courses/280031/files/24893968?wrap=1)

**Learning Goals**

1. Learn how to write a C program that follows the given specifications.
2. Learn how C address the problem of creating complex data structures without the benefit of object-oriented programming

**Reading**

* begin with this reading to understand the data structure
* We’re using the Head struct variant on page 26. Our head struct is called LINKED\_LIST and only contains a pointer to the first node
* [LinkedListBasics.pdf](https://canvas.wisc.edu/courses/280031/files/24839330?wrap=1)

**Compiling and Running**

* All of the work will be done in the file linked\_list.c
* A few functions have been written for you to help you get started
* Complete the remaining function definitions
* A simple main function has been provided that performs one simple test for each function. Make sure you expand main to write more comprehensive tests.

gcc -g linked\_list.c -Wall  
./a.out

**Turn in**

Upload your linked\_list.c file to Canvas before 11:59:00 pm.  You my upload your work as many times as you wish.  We will grade the last file submitted.

**Style**

Please follow the style guide linked on the Canvas Homepage.

**Logistics**

1. Do all of your work on the CSL instructional machines.
2. All assignments in this course are graded only on CS departmental machines running Linux Operating System. It is your responsibility to make sure that your code runs on these machines correctly.
3. The project is due at exactly 11:59:00 pm. You may upload your file as many times as you wish. We will grade the last file submitted. At 11:59:00 pm, you will no longer be able to submit your work.
4. We will be dropping one project score.

**Debugging advice**

1. Write the comments first.
2. Write your code in very small sections at a time and verify that it compiles before continuing.
3. Fix the compiler error messages from the top down. The first error may cause other compiler error messages to generate from the correct code later in your file.  So, scroll up and address the top error message first.
4. Add print statements – lots of print statements.
5. Use gdb to trace through your project.
6. Ask questions about cryptic error messages. Post a screenshot of the error message on Piazza as a public post if you do not need to share any code. Post privately if you do need to share your code.
7. Piazza and the TAs are not debuggers. These resources are there to teach you how to debug your own code.