

# CSC/DSCI 2720 Data Structure

## Lab 7

**Due: 02/25/2024 @ 11:59PM**

### Requirements:

(Failure to follow the requirements will result in a score of Zero.)

1. You may use whatever IDEs / editors you like, but you must submit your responses on iCollege as **.py files**.
2. Your submissions will work exactly as required.
3. Make sure you submit a file that compiles.
4. Your submission will show an output. Should you receive a Zero for no output shown do not bother to email with “but the logic is perfect”.
5. Your program’s output must exactly match the specs (design, style) given here for each problem to pass the test cases.
6. Design refers to how well your code is written (i.e., is it clear, efficient, and elegant), while Style refers to the readability of your code (correct indentation, good variable names). Add comments to have necessary explanations for your program.
7. Add a “heading” at the very beginning of your .java files as follow:  
*Your Name*  
*CSc 2720 Lab #N*  
*Lab time: put your lab time here*  
*Due time: put the due date here*
8. \*\*\* If you used any website/online resources as reference, please cite in your comments what website did you use for studying the lab content. Also, you are supposed to make changes to the resource you use to make it your own version. Otherwise, your submission will be considered as plagiarism. \*\*\*

This lab we will explore on ways to do create and run operations on a Linked List data-structure.

[100 points] Create a Linked List data structure by writing your own Linked List class which contains a node class. Just the way we explored it in the Lecture, your node class will house the data (integer in this case) and a pointer to the next node element. Populate your linked list with the following integers and print it (you need to also print the commas).

**50, 11, 33, 21, 40, 71**

Delete  $N^{\text{th}}$  node from the end of the linked list and print the linked list after deletion. Here  $N = 2$

Below is the expected output after deleting the second last element. **50, 11, 33, 21, 71**

ATTN: Note here we do not know the length of the list.

Complete the above deletion operation without calculating the length of the list. Your solution should only make a single pass through the linked list, adhering to  $O(n)$  time complexity overall and  $O(1)$  space complexity.

Hint: Maintain two pointers: a 'Fast' Pointer and a 'Slow' pointer. Initialize both pointers to a dummy node which points to the head of the list. Then starting a counter from zero, move the 'fast' pointer two places forward, to maintain a gap of two between the fast and slow pointers and then move both in tandem. Finally, when the fast pointer reaches the end of the list, the slow pointer will be at the third last node. You can now delete the second last node.

Very Very Important:

1. (1) Your code should be well commented which explains all the steps you are performing to solve the problem. **A submission without code comments will immediately be deducted 15 points!**
2. (2) As a comment in your code, please write your test-cases on how you would test your solution assumptions and hence your code.  
**A submission without test cases will immediately be deducted 15 points!** Example of cases to be tested for are like: What if the array input which is expected does not exist - that is, input is a null. How should your code handle such a situation? Maybe output some message like "Null input case, so no output"? What if the length of the array is one? ... so on and so forth.

Please Remember: Although, written as comments - You will address your test cases in the form of code and not prose :)