

# Lab 2 – Linear Data Structures

**Release: 15 Sep 2023 (Fri, Week 5)**

**Due: Online submission in eLearn by Sunday, 24 Sep 2023, 11pm**

## Instructions:

1. This is an individual lab. You will be awarded class participation points for the labs component based on the completion of submission, not on the efficiency of solution (you are encouraged to explore the most efficient approach for your own learning purposes)
2. You can discuss the general approach to solve the questions but you should not be sharing your codes with anyone.
3. Unless specifically stated in the question, you are allowed to create additional methods in the classes.
4. Please ensure that all your codes can be compiled.
5. Multiple submissions are allowed but only the most recent submission will be kept in the system as the final submission.
6. Please zip up your submission as named the zip file as <StudentName>.zip.
7. The deadline is final, late submissions will not be accepted. Students with valid reasons may appeal and receive a penalized mark, subject to the teaching team's discretion.

## Q1 – Swapping Elements in a Single Linked List

An implementation of a Single Linked List has been given in SinglyLinkedList.java. Study the codes and complete the following requirements in SinglyLinkedList.java to extend the functionality of the linked list. Please ensure the head and tail nodes are set accordingly if they are impacted by the method. Compile and run SinglyLinkedListTest.java to test your implementation.

- Write a method named **swap** that swap the sequence of all the elements whereby the biggest value will be swapped with the lowest value, the second biggest value will be swapped with the second lowest value....
- You can **assume all the values are unique**.
- You **should not be** making changes (e.g. adding new getter methods) to the Node class within SinglyLinkedList or to update the values of the element object within Node class directly.
- You should try to include more test cases (e.g. using a sequence of (1 to 1000) or (1 to 10000) to see how your code will respond to a bigger input size

Before Swap :

Linked List : 3 2 1

First Element : 3

Last Element : 1

After Swap :

Linked List : 1 2 3

First Element : 1

Last Element : 3

-----  
Before Swap :

Linked List : 5 1 2 3

First Element : 5

Last Element : 3

After Swap :

Linked List : 1 5 3 2

First Element : 1

Last Element : 2

-----  
Before Swap :

Linked List : 5 3 4 1 2

First Element : 5

Last Element : 2

After Swap :

Linked List : 1 3 2 5 4

First Element : 1

Last Element : 4