**CST-201**

**Project 2: Singly-Linked List**

The purpose of this assignment is to assess your ability to:

* Implement sequential search algorithms for linked list structures
* Implement sequential abstract data types using linked data
* Analyze and compare algorithms for efficiency using Big-O notation

For this assignment, you will implement a singly-linked node class. Use your singly-linked node to implement a singly-linked list class that maintains its elements in ascending order.

The **SinglyLinkedList** class is defined by the following data:

* A node pointer to the front and the tail of the list

Implement the following methods in your class:

* A default constructor list<T> myList
* A copy constructor list<T> myList(aList)
* Access to first element myList.front()
* Access to last element myList.back()
* Insert value myList.insert(val)
* Remove value at front myList.pop\_front()
* Remove value at tail myList.pop\_back()
* Determine if empty myList.empty()
* Return # of elements myList.size()
* Reverse order of elements in list myList.reverse()
* Merge with another ordered list myList.merge(aList)

Write a thorough test program for this class. Create a Loom video in which you run your program and offer a brief explanation of your implementation. Time analysis should be included. Your video should not exceed 4 minutes in length. Play your video before you submit to ensure that the quality is acceptable. Videos with sound issues will not be considered.

Submit the following in LoudCloud:

* All code in a single zipped file
* A document that contains
  + Your name
  + A statement that the submitted work is your own
  + A link to your Loom video