# COMP 200: Data Structures and Algorithms
# Fall 2020
# Lab 6: Linked Lists

**Task 1:** Implement the _DoublyLinkedBase class from page 274 of the book and PositionalList class from page 282-284 of the textbook/

**Task 2:** Modify the _DoublyLinkedBase class to include a reverse method that reverses the order of the list, yet without creating or destroying any nodes.

**Task 3:** Modify the PositionalList class to support a method swap(p, q) that causes the underlying nodes referenced by positions p and q to be exchanged for each other. Relink the existing nodes; do not create any new nodes.

**Task 4:** There is a simple, but inefficient, algorithm, called bubble-sort, for sorting a list L of n comparable elements. This algorithm scans the list n−1 times, where, in each scan, the algorithm compares the current element with the next one and swaps them if they are out of order. Implement a bubble sort function that takes a positional list L as a parameter. What is the running time of this algorithm, assuming the positional list is implemented with a doubly linked list?