# CSC 212 Tutorial #4
# List & DLL

## Problem 1

**Method**: reverse( ): **requires**: none. **input**: none. **results**: elements of the list will be stored in reverse order. **output**: none.

**Example 1.1.** *Given the list:* $20, 11, 44, 33, 50, 44$, reverse() *results in:* $44, 50, 33, 44, 11, 20$

1. Write the reverse method as an implementer of the *LinkedList ADT*
2. Write the reverse method as a user of the *List ADT*

## Problem 2

Write the method *circularLeftShift*, user of *List ADT*, that takes as input a non-empty *List* list and an integer $n > 0$ and performs $n$ circular left shift of the list.

**Example 2.1.** *Given the list* $l : A, B, C, D, E$, circularShiftLeft(l, 1) *results in* $B, C, D, E, A$, circularShiftLeft(l, 2) *results in* $C, D, E, A, B$.

## Problem 3

Write the method *removeBetween*, member of the class *DoubleLinkedList*. The method takes two elements $e_1$ and $e_1$, and removes all the elements between the two elements ($e_1$ and $e_2$ not included). If $e_1$ or $e_2$ or both doesn't exist, no element will be removed. You can assume the elements to be unique, and that $e_1 \neq e_2$. **Do not call any methods and do not use any auxiliary data structures**. The method signature is: `public void removeBetween(T e1, T e2)`.

**Example 3.1.** *Given the list:* $A \leftrightarrow B \leftrightarrow C \leftrightarrow D \leftrightarrow E \leftrightarrow F$, removeBetween('B', 'E') *results in:* $A \leftrightarrow B \leftrightarrow E \leftrightarrow F$.

## Problem 4

Write the method *reverseCopy*, user of *DoubleLinkedList*, which copies the elements of $l1$ to $l2$ in reverse order. The list $l1$ must not change. Assume that $l2$ is empty. The method signature is `public static <T> void reverseCopy(DoubleLinkedList<T> l1, DoubleLinkedList<T> l2)`.

**Example 4.1.** *If* $l1 : A \leftrightarrow B \leftrightarrow C \leftrightarrow D$*, then calling* reverseCopy(l1, l2) *results in* $l2 : D \leftrightarrow C \leftrightarrow B \leftrightarrow A$.