**CSCI 230 – Data Structures and Algorithms
Fall 2019
Lab 3: Linked List, Stack, Queue**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercise 1**

Consider the double linked list implementation from OpenDSA, where Link is defined as:
**class** Link<E> { *// Doubly linked list node*

 **private** E e; *// Value for this node*

 **private** Link<E> n; *// Pointer to next node in list*

 **private** Link<E> p; *// Pointer to previous node*

 *// Constructors*

 Link(E it, Link<E> inp, Link<E> inn) { e = it; p = inp; n = inn; }

 Link(Link<E> inp, Link<E> inn) { p = inp; n = inn; }

 *// Get and set methods for the data members*

 **public** E element() { **return** e; } *// Return the value*

 **public** E setElement(E it) { **return** e = it; } *// Set element value*

 **public** Link<E> next() { **return** n; } *// Return next link*

 **public** Link<E> setNext(Link<E> nextval) { **return** n = nextval; } *// Set next link*

 **public** Link<E> prev() { **return** p; } *// Return prev link*

 **public** Link<E> setPrev(Link<E> prevval) { **return** p = prevval; } *// Set prev link*

}

Write a printReverse method that prints the list starting from the tail. Use commas to separate the elements of the list.

**Exercise 2:**

Now consider a single linked list with link defined as:

**class** Link<E> { *// Singly linked list node class*

 **private** E e; *// Value for this node*

 **private** Link<E> n; *// Point to next node in list*

 *// Constructors*

 Link(E it, Link<E> inn) { e = it; n = inn; }

 Link(Link<E> inn) { e = **null**; n = inn; }

 E element() { **return** e; } *// Return the value*

 E setElement(E it) { **return** e = it; } *// Set element value*

 Link<E> next() { **return** n; } *// Return next link*

 Link<E> setNext(Link<E> inn) { **return** n = inn; } *// Set next link*

}

Write a printReverse method that prints the list starting from the tail. Use commas to separate the elements of the list.

**Exercise 3:**

Write a removeDuplicates method to remove duplicate elements from a Double Linked List.