**King Fahd University of Petroleum & Minerals**

**College of Computer Science and Engineering**

**Information and Computer Science Department**

**ICS 202 – Data Structures**

# Linked Lists-Part II

**Objectives**

The objective of this lab is to design, implement and use Doubly Linked Lists

**Outcomes**

After completing this Lab, students are expected to:

* Design Linked List classes.
* Implement Linked List classes (Doubly).
* Use Doubly Linked Lists in developing applications.

**Lab Exercises**

1. Download, compile and execute the programs related to Doubly linked lists. (class **DLList**).

2. Write a method **public void printReverse()** that prints the elements of a doubly linked list in reverse.

3. Write a method **public void delete7()** in the DLL<T> class which deletes the 7th element from the list starting from the head (which is the 1st element). Note that if you reach the end (tail) then you have to reverse the direction of counting. In the *main*() method of the test class, create a randomly generated Doubly-Linked list of 10 Integers**.** Next, call the*delete7()* method and print the lists iteratively until the list becomes empty.Make sure to print the list after each deletion.

For example, your list initially could be: [ 3 1 2 5 8 7 9 0].

After deleting 7th element: [ 3 1 2 5 8 7 **9** 0] => [ 3 1 2 5 8 7 0].

After deleting 7th element again; [ 3 1 2 5 8 7 **0**] => [ 3 1 2 5 8 7].

After deleting 7th element again (counting in the same direction, then moving reverse),

[3 1 2 5 **8** 7] => [3 1 2 5 7]

[3 1 **2** 5 7] => [3 1 5 7]

4. Write a method **public void insertAlternate(DLL<T> newList)** that takes a doubly linked list newList as a parameter and inserts the elements of newList into the invocating list alternately. For example, if the invocating list is [1 2 3 4] and the newList is [5 6 7 8], then after the method call the invocating list is [1 5 2 6 3 7 4 8]. Assume that the lengths of the invocating (calling) list and newList are the same. Write a message if they are not the same.