**Assignment 9**

Section 7.3 talks about linked list with a header node and a trailer node. These nodes are not part of the list; but their presence makes implementation simpler. Create a ***sorted doubly*** linked list called *DLSortedList<T>* with such a header node and a trailer node.

1. Create methods: addHeader(T header) and addTrailer(T trailer) (These methods allow adding a header and a trailer first before using “add” method to add other items.)
2. Implement the following methods:

*add, remove, and toString*

(Note that with the header and trailer implementations of add and remove methods are simpler.)

1. The list implements Iterator interface
2. Create copy constructor:

*public DLSortedList(DLSortedList<T> anotherList){ … }*

**(Make sure that deep copies are created.)**

1. Create a Student class as follows:

*class Student implements (some interfaces as needed){*

*Name sName;*

*double gpa;*

*/\* constructors and needed methods are at your discretion \*/*

*}*

where type Name is defined as follows:

*class Name{*

*String first, last;*

*/\* You add what you need. \*/*

*}*

Then, add objects of Student to the list you created, and test the methods you’ve implemented (comparison between two Student objects can be based on their gpa’s, and then their names).