Bowen Boyd

Tyler Allen

CS 136L

April 18, 2017

**Lab 09 - Doubly Linked List**

**1.Problem Statement**

In this lab we were to modify the linked list object in java so that it is capable of moving forward and backwards between nodes. We were then suppose to devise a Junit Test to check our program. In the spirit of test driven design were to write the test and the program at the same time.

**Requirements**

• You must be able to retrieve an iterator object with listIterator(int)

• You must be able to add to an empty list using an iterator

• You must be able to read the first element of a list using an iterator

• You must be able to move forward and backwards through your list using an iterator

• You must be able to add arbitrary elements using an iterator

• You must be able to read arbitrary elements using an iterator

• You must be able to remove elements using an iterator

• You must be able to replace a value in the list using an iterator

• You must be able to determine if you are at the beginning or end of a list using an iterator

• You must be able to determine your current index using an iterator

• You must be able to retrieve iterator objects at a non-zero index from listIterator(int)

• You must be able to correctly retrieve the size of the list after adding and removing elements correctly

• You must be able to use addAll(Collection) to add every item in an existing collection. • You must be able to read from the list by index

• You must be able to remove from the list by index

• You must be able to add to the list by index

• You must be able to replace an element by index

• Trying to iterate past the end of the list, or previous to the beginning should throw a NoSuchElementException

**2.Planning**

The DoublyLinkedList file will contain three different classes:

* DoublyLinkedList class

- Class containing the list

* DoublyLinkedListIterator class

- class which will manipulate the iterator

- directly add and change nodes in the list

* Node class

-Type of container used in the list

**3.Implementation and Planning**

**Screenshot of Run**

**4. Reflection**