



Linked List Lab

After finishing each part of the lab, copy your entire project and work on the copy for the next part!

Part 7: Add methods to add/remove items from the middle of a *MyLinkedList* list.

Add an `add` method to the *MyLinkedList* class

- Check for a valid insertion location; throw an *IndexOutOfBoundsException* if needed. Note that adding at index 0 is always valid.
- Traverse to *index-1*
- Copy the *next* reference for node *index-1* into the *next* reference for the new node.
- Set the *next* reference for node *index-1* so it points to the new node.
- Check if the *head* pointer should be adjusted (is the new node at index 0?).
- Check if the *tail* pointer should be adjusted (is the new node now the last node?).
- Be sure to increment the size counter.

Add a `remove` method to the *MyLinkedList* class

- Check for a valid insertion location; throw an *IndexOutOfBoundsException* if needed.
- Traverse to *index*, keeping a reference (*previous*) to the trailing node.
- Copy the *next* reference for node *index* into the *next* reference for the *previous* node.
- Check if the *head* pointer should be adjusted (was the deleted node at index 0?).
- Check if the *tail* pointer should be adjusted (was the deleted node the last node?).
- Be sure to decrement the size counter.

```
public class MyLinkedList<T>
-----
    void  add(int index, T data)    // insertion, not replacement, at position index
                                   //      throws IndexOutOfBoundsException
    T  remove(int index)          // throws IndexOutOfBoundsException
```

Write a *ListTester* class

- Write the class *ListTester*, containing a *main* method.
- Make a *MyLinkedList<Actor>* object.
- The 1st part of *main* loads data from a file (“actors7.txt”) into the linked list:
 - Read a name from the file & build an *Actor* object.
 - Add the *Actor* object to the linked list, using the *addFirst* method in your *MyLinkedList* object.
 - Read & add the second *Actor* to the list using the *addLast* method.
 - There are 8 names on the list. Repeat adding two items in a loop to process all eight names.
- The 2nd part of *main* prints data from the linked list:
 - Print the number of actors on the list on the first line of output using *MyLinkedList*’s *size* method.
 - Print the list using the *toString* method from the *MyLinkedList* object.

- The 3rd part of *main* manipulates & prints the list:
 - Read an integer & a name from the file and build an *Actor* object.
 - Add the *Actor* object to the linked list at the index given, using the *add(i, obj)* method in your *MyLinkedList*.
 - Read an integer from the file and delete the *Actor* object at that location using the *remove(i)* method.
 - Repeat adding an object & removing an object. [There are 2 add and 2 remove operations.]
 - Print the number of actors on the list using *MyLinkedList*'s *size* method.
 - Print the list using the *toString* method from the *MyLinkedList* object.

Sample Input (actors7.txt)

```
William Shatner
Leonard Nimoy
DeForest Kelley
Patrick Stewart
Jonathan Frakes
James Doohan
Walter Koenig
George Takei
8
LeVar Burton
1
3
Brent Spiner
5
```

Sample Output

```
8
Walter Koenig
Jonathan Frakes
DeForest Kelley
William Shatner
Leonard Nimoy
Patrick Stewart
James Doohan
George Takei
8
Walter Koenig
DeForest Kelley
William Shatner
Brent Spiner
Leonard Nimoy
James Doohan
George Takei
LeVar Burton
```