

**Lab Goal :** This lab was designed to teach you more about a linked list.

**Lab Description :** Write a single-linked linked list class. Each node in `SinglyLinkedList` has a data component and a reference to the next node only. Write and test all methods in `SinglyLinkedList`. Complete methods to add, remove, get, and search.

Any method that accepts an index as an input must throw an `IndexOutOfBoundsException` with a message that includes the offending index and the size of the list. Any method that is supposed to access or change the first or last elements should throw a `NoSuchElementException` when the list is empty.

### ListNode – stores a value and a reference to the next node

```
public class ListNode implements Linkable
{
    private Comparable listNodeValue;
    private ListNode nextListNode;

    public ListNode(){
        listNodeValue = null;
        nextListNode = null;
    }

    public ListNode(Comparable value, ListNode next){
        listNodeValue=value;
        nextListNode=next;
    }

    public Comparable getValue(){
        return listNodeValue;
    }

    public ListNode getNext(){
        return nextListNode;
    }

    public void setValue(Comparable value){
        listNodeValue = value;
    }

    public void setNext(Linkable next){
        nextListNode = (ListNode)next;
    }
}
```

### Sample Data :

```
[4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
[24, 75, 86, 37, 82, 94, 111, 82, 43]
[0, 4, 5, 2, 1, 4, 6]
[]
```

### Sample Output 1:

```
*****
*** Test add, addFirst, addLast, and toString ***
*****
LIST:: 4 5 6 7 8 9 10 11 12 13
LIST2:: 4 5 6 7 8 9 10 11 12 13
LIST3:: 20 4 5 6 7 8 9 10 11 12 13 40

*****
*** Test get, getFirst, getLast, size, contains, and equals ***
*****
list.get(2) == 6
```

```
list.get(100) should throw an IndexOutOfBoundsException.
Your method threw: java.lang.IndexOutOfBoundsException: Index: 100, Size: 10
```

```

list.get(-1) should throw an IndexOutOfBoundsException.
    Your method threw:  java.lang.IndexOutOfBoundsException: Index: -1, Size: 10

Testing getFirst and getLast:
    list.getFirst() == 4
    list.getLast() == 13

Test size and contains:
    list.size() == 10
    list.contains(8) == true
    list.contains(14) == false

Comparing list to itself:
    list.equals(list)? == true
Comparing list to an identical list:
    list.equals(list2)? == true
Comparing list to a list with different contents:
    list.equals(list3)? == false

*****
*** Test set, remove(int), remove(obj), removeFirst, and removeLast on list ***
*****
list2:: 4 5 6 7 8 9 10 11 12 13

set(2, 100) returned 6
    list2 after set(2, 100): 4 5 100 7 8 9 10 11 12 13

list2.set(100, 2) should throw an IndexOutOfBoundsException.
    Your method threw:  java.lang.IndexOutOfBoundsException: Index: 100, Size: 10

Testing remove(spot) on the first item:
    remove(0) returned 4
    list2 after remove(0):  5 100 7 8 9 10 11 12 13

Testing remove(spot) on the last item:
    remove( list.size()-1 ) returned 13
    list2 after remove(list.size()-1):  5 100 7 8 9 10 11 12

Testing remove(spot) on a middle item:
    remove(2) returned 7
    list2 after remove(2):  5 100 8 9 10 11 12

Testing remove(spot) on a invalid spot:
    list2.remove(100) should throw an IndexOutOfBoundsException.
    Your method threw:  java.lang.IndexOutOfBoundsException: Index: 100, Size: 7

Testing remove(obj) on the first item:
    remove(Integer.valueOf(5)) returned true
    list2 after remove(Integer.valueOf(5)):  100 8 9 10 11 12

Testing remove(obj) on the last item:
    remove(Integer.valueOf(12)) returned true
    list2 after remove(Integer.valueOf(12)):  100 8 9 10 11

Testing remove(obj) on an item that IS in the list:
    remove(Integer.valueOf(8)) returned true
    list2 after remove(Integer.valueOf(8)):  100 9 10 11

Testing remove(obj) on an item that is NOT in the list:
    remove(Integer.valueOf(14)) returned false
    list2 after remove(Integer.valueOf(14)):  100 9 10 11

Testing removeFirst and removeLast:
    removeFirst() returned 100
    list2 after removeFirst():  9 10 11
    removeLast() returned 11
    list2 after removeLast():  9 10

*****
*** Test clear and that NoSuchElementException are thrown when appropriate ***
*****
list3.clear()
    list3 after clear() is called:

list3.getFirst() should throw an NoSuchElementException.
    Your method threw:  java.util.NoSuchElementException

```

```
list3.getLast() should throw an NoSuchElementException.
Your method threw:  java.util.NoSuchElementException

list3.removeFirst() should throw an NoSuchElementException.
Your method threw:  java.util.NoSuchElementException

list3.removeLast() should throw an NoSuchElementException.
Your method threw:  java.util.NoSuchElementException
```

```
*****
*** Test List Utility methods ***
*****
      SUM:: 85.0
    AVERAGE:: 8.5
    SMALLEST:: 4
    LARGEST:: 13
```

## Sample Output 2:

```
*****
*** Test add, addFirst, addLast, and toString ***
*****
LIST:: 24 75 86 37 82 94 111 82 43
LIST2:: 24 75 86 37 82 94 111 82 43
LIST3:: 20 24 75 86 37 82 94 111 82 43 40

*****
*** Test get, getFirst, getLast, size, contains, and equals ***
*****
      list.get(2) == 86

list.get(100) should throw an IndexOutOfBoundsException.
Your method threw:  java.lang.IndexOutOfBoundsException: Index: 100, Size: 9

list.get(-1) should throw an IndexOutOfBoundsException.
Your method threw:  java.lang.IndexOutOfBoundsException: Index: -1, Size: 9

Testing getFirst and getLast:
list.getFirst() == 24
list.getLast() == 43

Test size and contains:
list.size() == 9
list.contains(37) == true
list.contains(100) == false

Comparing list to itself:
list.equals(list)? == true
Comparing list to an identical list:
list.equals(list2)? == true
Comparing list to a list with different contents:
list.equals(list3)? == false

*****
*** Test set, remove(int), remove(obj), removeFirst, and removeLast on list ***
*****
list2:: 24 75 86 37 82 94 111 82 43

set(2, 100) returned 86
list2 after set(2, 100): 24 75 100 37 82 94 111 82 43

list2.set(100, 2) should throw an IndexOutOfBoundsException.
Your method threw:  java.lang.IndexOutOfBoundsException: Index: 100, Size: 9

Testing remove(spot) on the first item:
remove(0) returned 24
list2 after remove(0): 75 100 37 82 94 111 82 43

Testing remove(spot) on the last item:
remove( list.size()-1 ) returned 43
list2 after remove(list.size()-1): 75 100 37 82 94 111 82

Testing remove(spot) on a middle item:
remove(2) returned 37
list2 after remove(2): 75 100 82 94 111 82
```

```

Testing remove(spot) on a invalid spot:
    list2.remove(100) should throw an IndexOutOfBoundsException.
    Your method threw:  java.lang.IndexOutOfBoundsException: Index: 100, Size: 6

Testing remove(obj) on the first item:
    remove(Integer.valueOf(75)) returned true
    list2 after remove(Integer.valueOf(75)):  100 82 94 111 82

Testing remove(obj) on the last item:
    remove(Integer.valueOf(82)) returned true
    list2 after remove(Integer.valueOf(82)):  100 94 111 82

Testing remove(obj) on an item that IS in the list:
    remove(Integer.valueOf(37)) returned false
    list2 after remove(Integer.valueOf(37)):  100 94 111 82

Testing remove(obj) on an item that is NOT in the list:
    remove(Integer.valueOf(100)) returned true
    list2 after remove(Integer.valueOf(100)):  94 111 82

Testing removeFirst and removeLast:
    removeFirst() returned 94
    list2 after removeFirst():  111 82
    removeLast() returned 82
    list2 after removeLast():  111

*****
*** Test clear and that NoSuchElementException are thrown when appropriate ***
*****
list3.clear()
    list3 after clear() is called:

list3.getFirst() should throw an NoSuchElementException.
    Your method threw:  java.util.NoSuchElementException

list3.getLast() should throw an NoSuchElementException.
    Your method threw:  java.util.NoSuchElementException

list3.removeFirst() should throw an NoSuchElementException.
    Your method threw:  java.util.NoSuchElementException

list3.removeLast() should throw an NoSuchElementException.
    Your method threw:  java.util.NoSuchElementException

*****
*** Test List Utility methods ***
*****
    SUM:: 634.0
    AVERAGE:: 70.44444444444444
    SMALLEST:: 24
    LARGEST:: 111

```

### Sample Output 3:

```

*****
*** Test add, addFirst, addLast, and toString ***
*****
    LIST:: 0 4 5 2 1 4 6
    LIST2:: 0 4 5 2 1 4 6
    LIST3:: 20 0 4 5 2 1 4 6 40

*****
*** Test get, getFirst, getLast, size, contains, and equals ***
*****
    list.get(2) == 5

list.get(100) should throw an IndexOutOfBoundsException.
    Your method threw:  java.lang.IndexOutOfBoundsException: Index: 100, Size: 7

list.get(-1) should throw an IndexOutOfBoundsException.
    Your method threw:  java.lang.IndexOutOfBoundsException: Index: -1, Size: 7

Testing getFirst and getLast:
    list.getFirst() == 0
    list.getLast() == 6

```

```

Test size and contains:
    list.size() == 7
    list.contains(6) == true
    list.contains(7) == false

Comparing list to itself:
    list.equals(list)? == true
Comparing list to an identical list:
    list.equals(list2)? == true
Comparing list to a list with different contents:
    list.equals(list3)? == false

*****
*** Test set, remove(int), remove(obj), removeFirst, and removeLast on list ***
*****
list2:: 0 4 5 2 1 4 6

set(2, 100) returned 5
    list2 after set(2, 100): 0 4 100 2 1 4 6

list2.set(100, 2) should throw an IndexOutOfBoundsException.
    Your method threw: java.lang.IndexOutOfBoundsException: Index: 100, Size: 7

Testing remove(spot) on the first item:
    remove(0) returned 0
    list2 after remove(0): 4 100 2 1 4 6

Testing remove(spot) on the last item:
    remove( list.size()-1 ) returned 6
    list2 after remove(list.size()-1): 4 100 2 1 4

Testing remove(spot) on a middle item:
    remove(2) returned 2
    list2 after remove(2): 4 100 1 4

Testing remove(spot) on a invalid spot:
    list2.remove(100) should throw an IndexOutOfBoundsException.
    Your method threw: java.lang.IndexOutOfBoundsException: Index: 100, Size: 4

Testing remove(obj) on the first item:
    remove(Integer.valueOf(4)) returned true
    list2 after remove(Integer.valueOf(4)): 100 1 4

Testing remove(obj) on the last item:
    remove(Integer.valueOf(4)) returned true
    list2 after remove(Integer.valueOf(4)): 100 1

Testing remove(obj) on an item that IS in the list:
    remove(Integer.valueOf(6)) returned false
    list2 after remove(Integer.valueOf(6)): 100 1

Testing remove(obj) on an item that is NOT in the list:
    remove(Integer.valueOf(7)) returned false
    list2 after remove(Integer.valueOf(7)): 100 1

Testing removeFirst and removeLast:
    removeFirst() returned 100
    list2 after removeFirst(): 1
    removeLast() returned 1
    list2 after removeLast():

*****
*** Test clear and that NoSuchElementException are thrown when appropriate ***
*****
list3.clear()
    list3 after clear() is called:

list3.getFirst() should throw an NoSuchElementException.
    Your method threw: java.util.NoSuchElementException

list3.getLast() should throw an NoSuchElementException.
    Your method threw: java.util.NoSuchElementException

list3.removeFirst() should throw an NoSuchElementException.
    Your method threw: java.util.NoSuchElementException

list3.removeLast() should throw an NoSuchElementException.
    Your method threw: java.util.NoSuchElementException

```

```
*****  
*** Test List Utility methods ***  
*****  
      SUM:: 22.0  
    AVERAGE:: 3.142857142857143  
  SMALLEST:: 0  
    LARGEST:: 6
```

## Sample Output 4:

**Empty List!**