Singly Linked List 3

Computer Science Department

Eastern Washington University

Yun Tian (Tony) Ph.D.

Review

- toString()
- Remove data element from linked list.
  - Many edge cases

Today

- List Iterator
- List with Dummy Node(aka head node)

LinkedList Class Implementation

```
public class LinkedList implements Iterable<Object> {
    private class Node {
        private Object data;
        private Node next;

        private Node( Object data, Node next ) {
            this.data=data;
            this.next = next;
        }
        private Node( Object data ) {
            this(data, null);
        }
        private Node() {} // Can we leave out this empty constructor?
    }//end of node

    private Node head;
    private int size;

    //………….to be continued on next page
```

List Iterator Implementation

```
public LinkedList() {
    this.head = null;
    this.size = 0;
}


@Override
public Iterator<Object> iterator() {
        return new MyLinkedListIterator(this.head);  //no dummy is right now
}
```

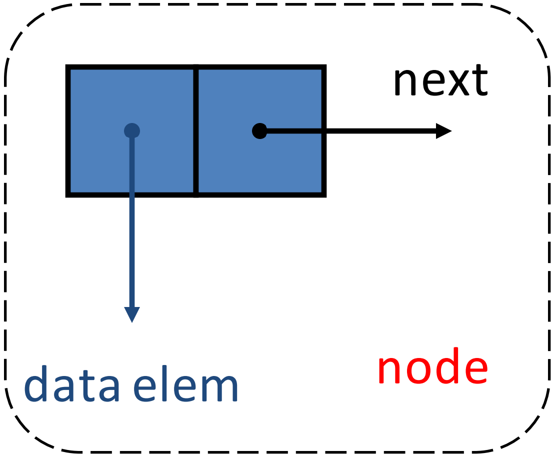
//First look at the demo of Linked List Iterator, posted on canvas Files->DemoCode-->D02

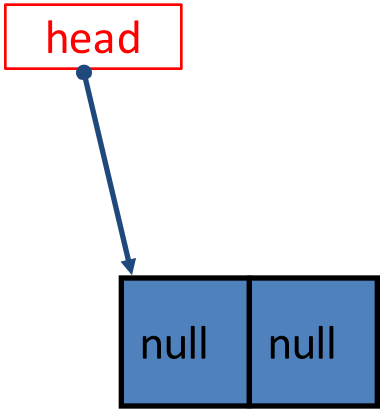
//………….**to be continued on next page**

List Iterator Implementation

```
public class MyLinkedListIterator implements Iterator<Object> { //inner class of LinkedList
      private Node cur;   //cur always points to the node we are about to access next!
      private int index;
      private MyLinkedListIterator ( Node start ) {
            this.cur = start ;
            this.index = 0;
       }
       public boolean hasNext() {
            return    this.cur != null    ;
       }
       public Object next() {
             if(hasNext()) {
                   Object data = cur.data;
                   cur = cur.next  ;
                   return data;
             }
             throw new NoSuchElementException();
       }
       public void remove() {
             throw new UnsupportedOperationException();  //you could implement this.
       }
}//end of MyLinkedLisIterator
```

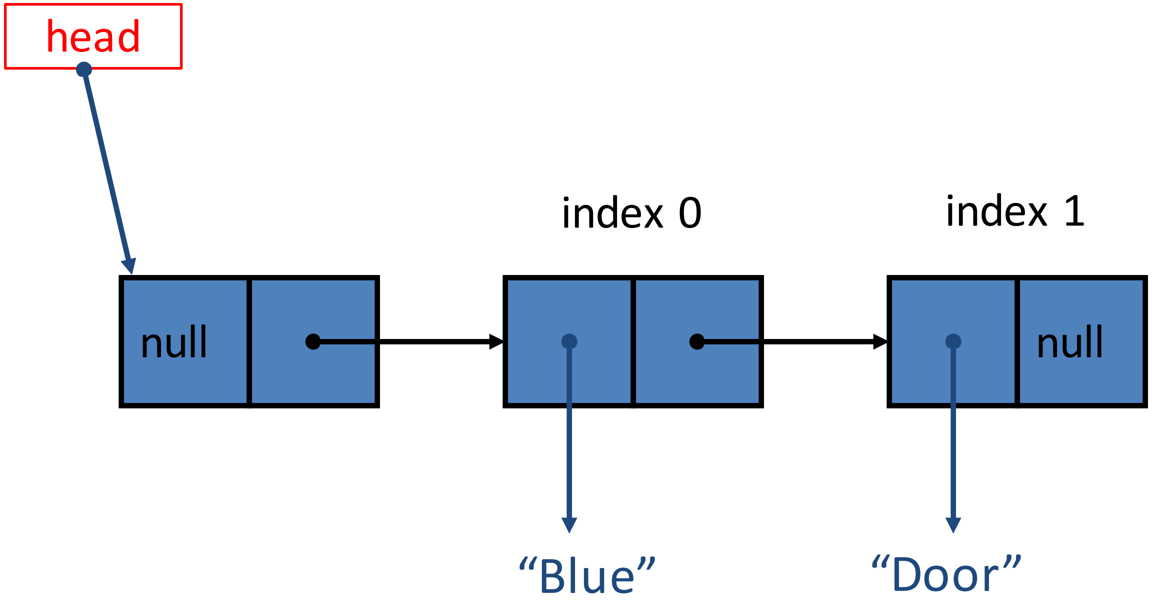
Singly Linked List with Dummy Node





With dummy node, this is what an **empty** LinkedList looks like. The first node is reserved, which does not hold data.

Singly Linked List with Dummy Node



With dummy node, this is what a LinkedList looks like. The first node is reserved, which does not hold data.

LinkedList with Dummy Node

```
public class MyLinkedList {

    private ListNode head;
    private int size;

    //inner class for ListNode
    private class ListNode {
        private Object data;
        private ListNode next;
        private ListNode(Object d) {
            this.data = d;
            this.next = null;
        }
        private ListNode() {
        }
    }

    public MyLinkedList() {
        this.head = new ListNode(null)_____;
        this.size = 0;
    }
    //…………
}
```

Remove data from List with Dummy Node

```
public boolean remove(Object o) { //remove the first occurrence of o
     for(ListNode ____prev = this.head____, ____cur = this.head.next____;
               cur != null; prev = cur, cur = cur.next){
          if (cur.data.equals(o)) { //should override equals method in your class Object
               prev.next = cur.next;
               this.size --;
               return true;
          }
     }//end for
     return false;
}
```

We don’t need to handle edge cases differently!!

Remove element from List Without Dummy Node

```
public boolean remove ( Object dataToRemove ) {
        if( isEmpty() | | dataToRemove == null )
            return false;
        Node cur = this.head, prev = null;
        while (___cur != null && ! cur.data.equals(dataToRemove)___){
            prev = cur;
            cur = cur.next;
        }
        if( ___cur == null___ ) // not existing
            return false;
        //edge case
        if(____prev == null___ )  //remove from front
        {
            this.head = this.head.next;
            this.size --;
            return true;
        }
        prev.next = cur.next;
        this.size --;
        return true;
}
```

Summary

- Implement Linkedlist Iterator
- LinkedList with a Dummy Node
- Remove(Object obj) is compared with its counterpart in a linkedlist without a dummy node.

Next class

- addOrdered() method
  - Add an element to a sorted linked list, in order to preserve the order.
- Use addOrdered() to Sort a Linked List
- Selection Sort Review