EASTERN WASHINGTON UNIVERSITY
start something big

Sorted Singly Linked List

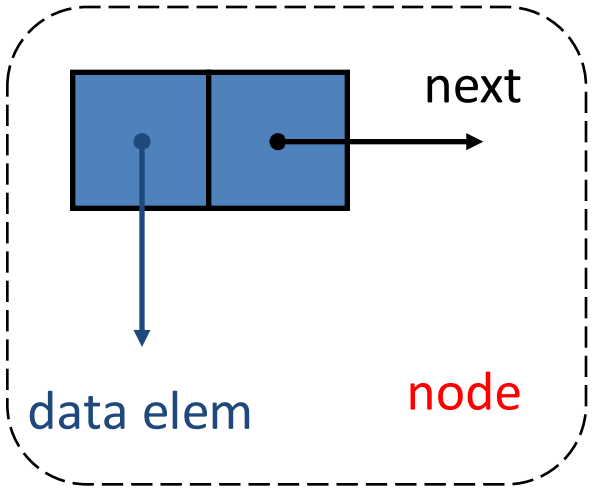
Computer Science Department
Eastern Washington University
Yun Tian (Tony) Ph.D.

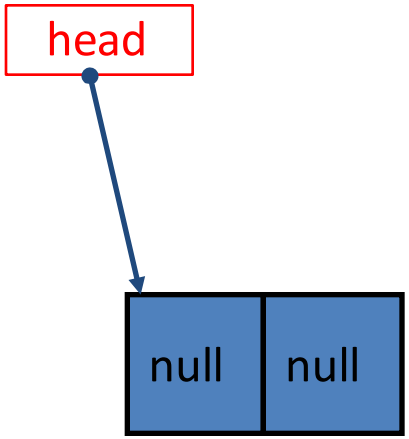
EASTERN WASHINGTON UNIVERSITY
start something big

Review

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

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.

EASTERN WASHINGTON UNIVERSITY
start something big

Today's Topics

- addOrdered() method
  - Add an element to a sorted linked list, in order to preserve its order.
- Use addOrdered() to Sort a Linked List
- Selection Sort On Array and LinkedList

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
```

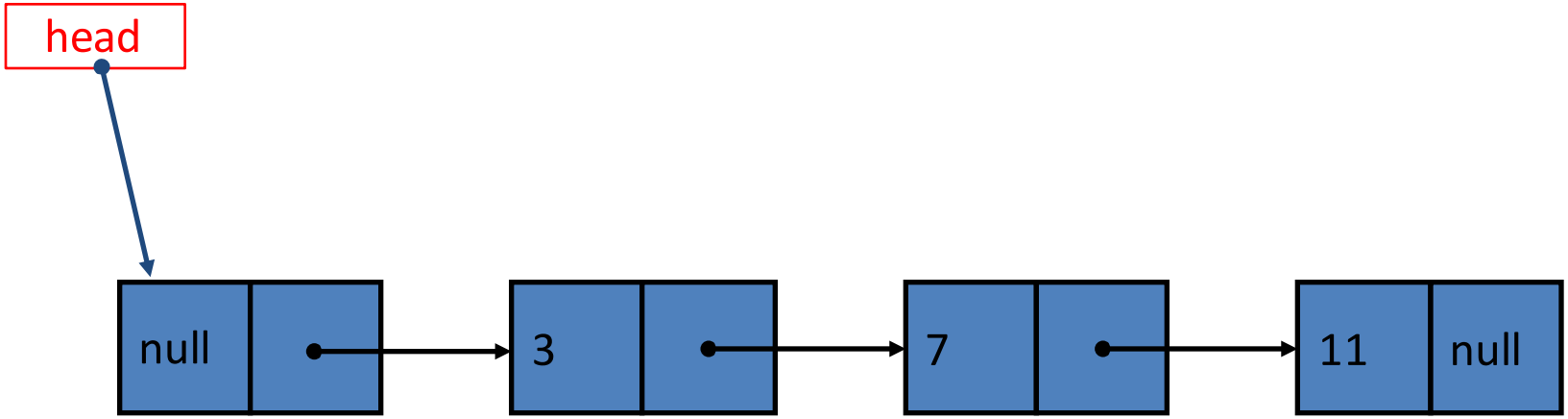
List Iterator Implementation

EASTERN WASHINGTON UNIVERSITY
start something big

```
public LinkedList() {
    this.head = new Node(null); //with dummy node
    this.size = 0;
}

@Override
public Iterator<Object> iterator() {
      return new MyLinkedListIterator(this.head.next);  //with a dummy right now
}
```

Sorted Singly Linked List



1) We like the list nodes to be sorted in an ascending order!
2) We can write an addOrdered(Comparable dataToAdd) method.
3) After we add an element, the list still keeps its ascending order.
4) We can create a sorted linkedlist using addOrdered method.

addOrdered Implementation

EASTERN WASHINGTON UNIVERSITY
start something big

```
public void addOrdered ( Comparable dataToAdd ) {
      // The precondition for the list that should be ordered before
      // The linked list has a Dummy Node
      Node cur,  prev;
      for( cur = this.head.next, prev = this.head;
                cur != null &&  ((Comparable) cur.data).compareTo(dataToAdd) < 0;  cur
= cur.next) {
                prev = cur ;
       }
        prev.next = new Node(dataToAdd,  cur) ;
       this.size ++;
}
```

Sort LinkedList with Dummy Node

EASTERN WASHINGTON UNIVERSITY
start something big

```
public void sort() { //flavor of insertion sort
        Node cur;
        LinkedList newList = new LinkedList();
        for(cur = this.head.next; cur != null; cur = cur.next)
                newList.addOrdered((Comparable)cur.data);
        this.head.next = newList.head.next;
}
public LinkedList sort2() { //flavor of insertion sort
        Node cur;
        LinkedList newList = new LinkedList();
        for(cur = this.head.next; cur != null; cur = cur.next)
                newList.addOrdered((Comparable)cur.data);

        return newList;
}
```

Selection Sort With Arrays

EASTERN WASHINGTON UNIVERSITY
start something big

```
public void selectSortArray( Comparable [] array ) {

    int smallest, cur, start;
    Comparable temp;
    for( start = 0; start < array.length - 1; start ++ ) {
        smallest = start;
        for( cur = start + 1; cur < array.length; cur ++ ) {
            if( array[cur].compareTo(array[smallest]) < 0 )
                smallest = cur ;
        }
        //swap smallest and and start;
        temp=array[start];
        array[start] = array[smallest];
        array[smallest] = temp;
    }//end of outer for
}
```

Selection Sort With LinkedList

EASTERN WASHINGTON UNIVERSITY
start something big

```
public void selectionSortList() {   //with dummy node
      if (this.size <= 1)    return;
      Node start, smallest, cur;
      Comparable temp;

      for(start = this.head.next ;    start.next != null ;  start = start.next ) {
                smallest = start;
                for( cur = start.next ;  cur != null; cur = cur.next) {
                      if ( ((Comparable)cur.data).compareTo((Comparable)smallest.data) < 0 )
                            smallest = cur ;
                }
                //swap
                temp = (Comparable)start.data;
                start.data = smallest.data;
                smallest.data = temp;
      }
}
```

EASTERN WASHINGTON UNIVERSITY
start something big

Summary

- addordered() method on Linked List
- Review Selection Sort on Array
- Selection Sort on Linked List with Dummy Node

Next class

- Circular Singly Linked List
- Circular Doubly Linked List