Lecture 14

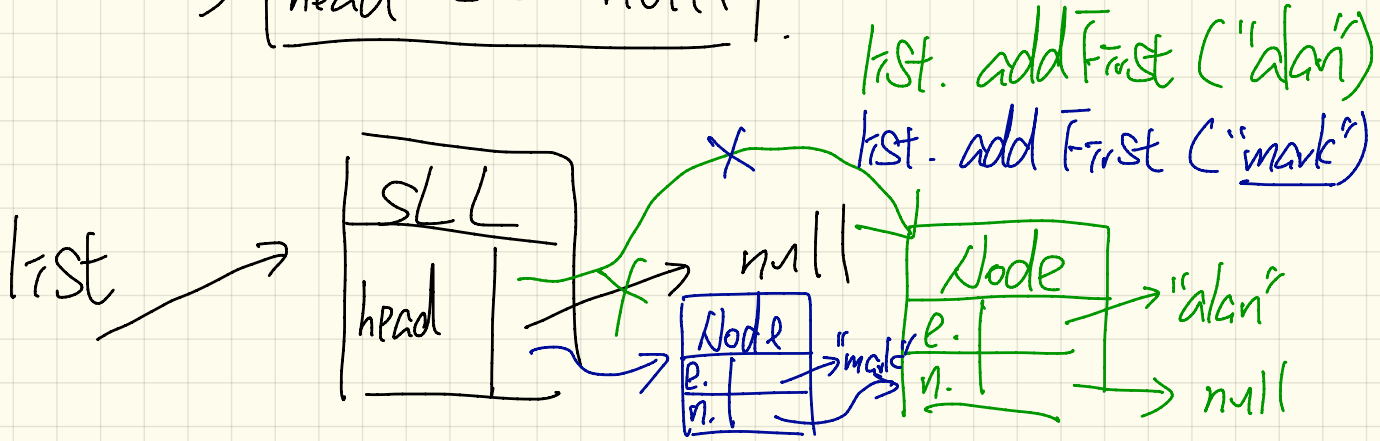
Tuesday Oct. 24

SLL empty

```
SLL list = new SLL();
```

↳ create an empty list

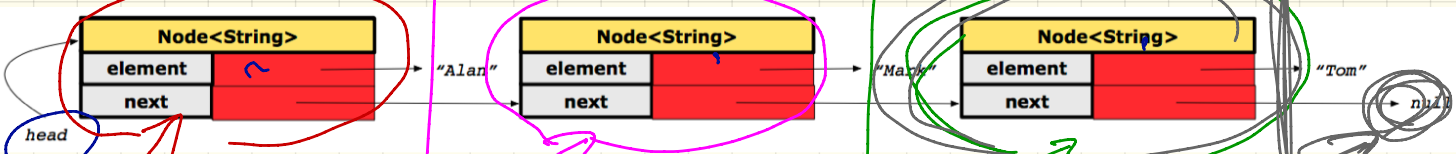
↳ `head == null`.



```
1  int getSize() {
2    int size = 0;
3    Node current = head;
4    while (current != null) {
5      /* exit when current == null */
6      current = current.getNext();
7      size ++;
8    }
9    return size;
10 }
```

① As soon as current becomes null, exit from the loop.

list.getSize() returns 3



current ①

current size becomes 1

current size becomes 2

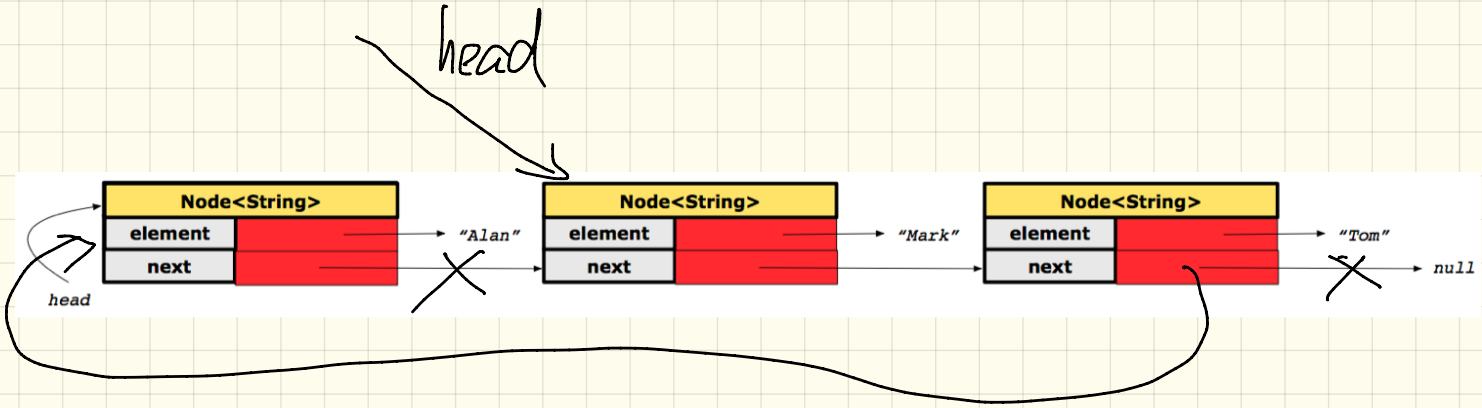
size becomes 3

current = current.getNext() ②

current = current.getNext() ③

current = current.getN()

null

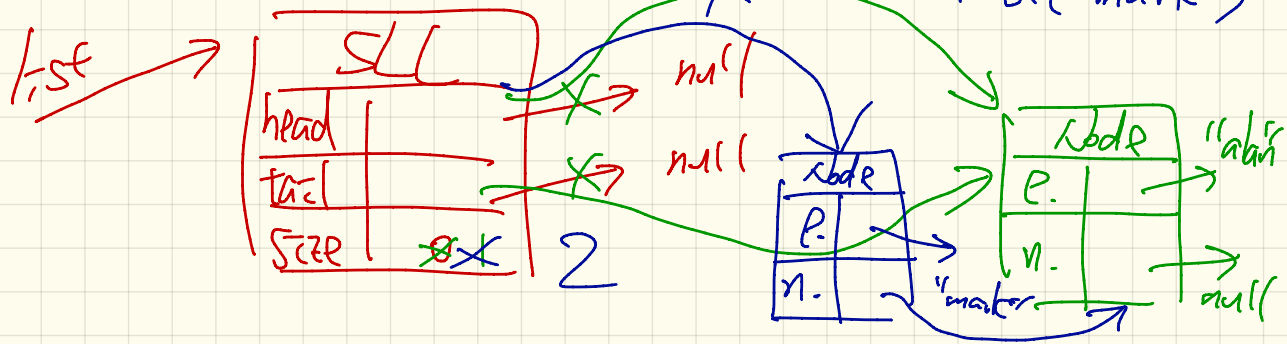


list.shiftToLeft()

```
1  Node getTail(Node head) {
2    Node current = head;
3    Node tail = null;
4    while (current != null) {
5      /* exit when current == null */
6      tail = current;
7      current = current.getNext();
8    }
9    return tail;
10 }
```

```
class SLL {
    Node head;
    Node tail;
    int size = 0;
    SLL() { head = null; tail = null; size = 0; }
}
SLL list = new SLL();
```

list.addFirst("alan")
list.addFirst("mark")



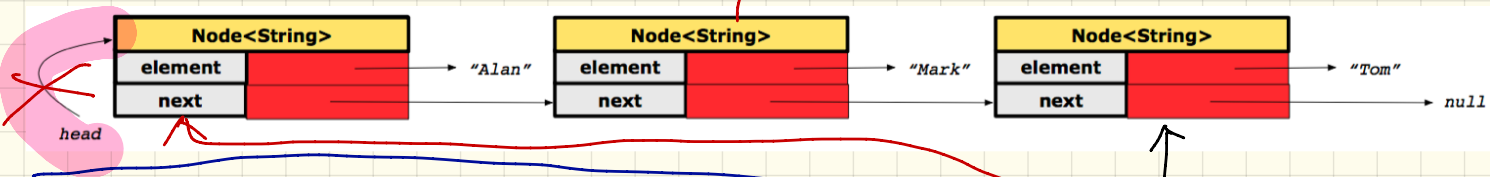
```
1  addFirst (String e) {
2    head = new Node(e, head);
3    if (size == 0) {
4      tail = head;
5    }
6    size ++;
7  }
```

Node nn = new Node(e, null);
nn.setNext(head);
head = nn;

list.addFirst("Jim")

special case for adding the very first node!

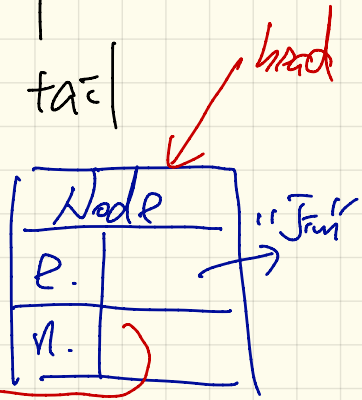
size == 0



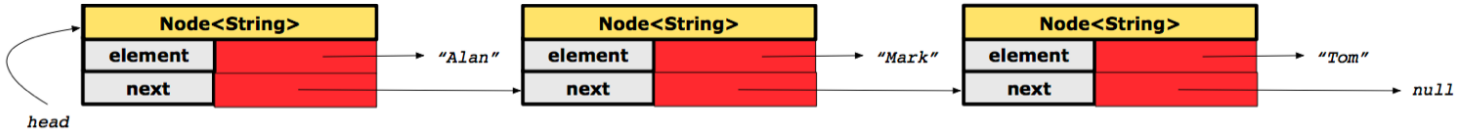

```
class Node {
    Node(String e, Node next) {
        this.element = e; this.next = next;
    }
}
```

head = new Node("Jim", head)

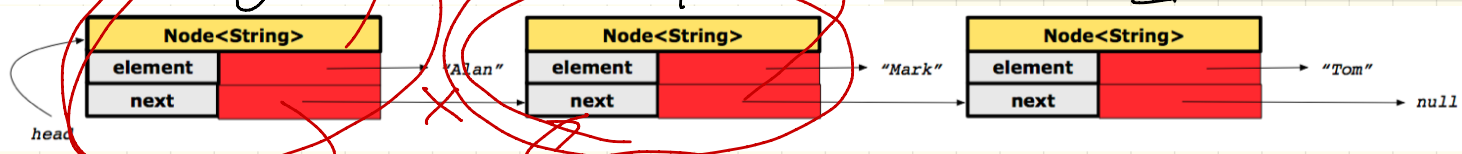
tail

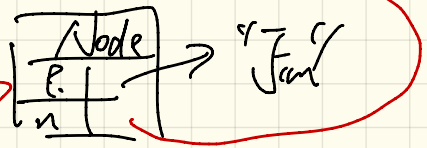


```
1  Node getNodeAt (int i) {
2    if (i < 0 || i >= size) {
3      throw IllegalArgumentException("Invalid Index");
4    }
5    else {
6      int index = 0;
7      Node current = head;
8      while (index < i) {  /* exit when index == i */
9        index ++;
10       /* current is set to node at index i
11        * last iteration: index incremented from i - 1 to i
12        */
13       current = current.getNext();
14     }
15     return current;
16   }
17 }
```



```
1  addAt (int i, String e) {
2    if (i < 0 || i >= size) {
3      throw IllegalArgumentException("Invalid Index.");
4    }
5    else {
6      if (i == 0) {
7        addFirst(e);
8      }
9      else {
10       Node nodeBefore = getNodeAt(i - 1);
11       newNode = new Node(e, nodeBefore.getNext());
12       nodeBefore.setNext(newNode);
13       size ++;
14     }
15   }
16 }
```





list.addAt(1, "Jean")

```
1  removeLast () {
2    if (size == 0) {
3      System.err.println("Empty List.");
4    }
5    else if (size == 1) {
6      removeFirst();
7    }
8    else {
9      Node secondLastNode = getNodeAt(size - 2);
10     secondLastNode.setNext(null);
11     tail = secondLastNode;
12     size --;
13   }
14 }
```