Last class:

- Complexity (finish)
  
  - Application to computer programs and analysis

Today:

- ADTs vs. Data Structures
- Chain of nodes a.k.a Linked Lists
- Practice with chains
ADTs vs. Data Structures

Data structure:

Abstract Data Type:
Arrays, Stacks and Queues

Recall upsides and downsides of arrays

Stack functionality

Queue functionality
Linked List – Chain of Linked Nodes

Conceptual Picture

What can they do?
Java Visibility Modifiers

public                   public class ArrayGSack
private                  private E[] items
protected               protected int numItems
int ID
(blank)

Providing package access

Access Levels:

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Class</th>
<th>Package</th>
<th>Subclass</th>
<th>World</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>protected</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(no modifier)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>private</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ListNode Class – Building Block

Interface:

```java
class ListNode<E> {
    public E getData()
    public void setData(E ob)
    public ListNode<E> getNext()
    public void setNext(ListNode<E> n)
}
```

Sample Java Implementation:

```java
(blank) class ListNode<E> {
    private E data;
    private ListNode<E> next;

    public ListNode(E d) { this(d, null); }
    public ListNode(E d, ListNode<E> n) {
        data = d;
        next = n;
    }
    public E getData() { return data; }
    public ListNode<E> getNext() { return next; }
    public void setData(E ob) { data = ob; }
    public void setNext(ListNode<E> n) { next = n; }
}
```

Why not make it public?
Working with chains – Example 1

Create a chain containing the strings “The”, “quick”, and “brown” in that order.
Working with chains – Example 2

Traversing a chain: Write a code fragment that counts the number of items in a given chain of nodes. Assume that head points to the first node in a chain of Integers.
Working with chains – Example 3

Write a code fragment that adds the number 68 to the end of the chain of nodes of Integers. Assume that head points to the first node in the chain.
Linked Lists – Pros and Cons

Advantages at a high level

Disadvantages at a high level
Working with chains: Removing a node

Write a code fragment to remove the third item from a chain of nodes. Assume head points to the first node in the chain. You may assume the chain has at least three items.