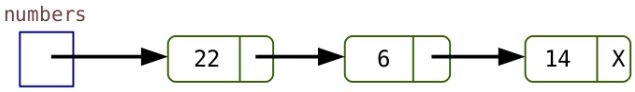
Model 1 Linked Lists

Linked structures "chain" elements using references. Each element of the list is called a ***node***.

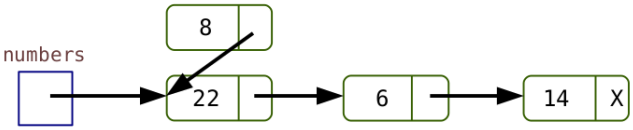
```
public class Node {

    private int value;
    private Node next;
    ...
}
```

```
Node node3 = new Node(14, null);
Node node2 = new Node(6, node3);
Node numbers = new Node(22, node2);
```

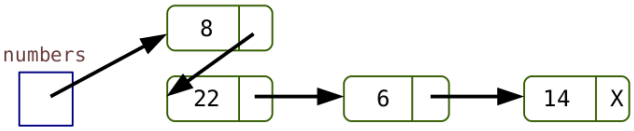


This organization allows fast insertions/deletions near the beginning. For example, to add 8:

```
Node temp = new Node(8, numbers);
```



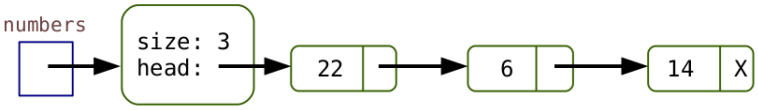

```
numbers = temp;
```



Instead of working with nodes directly, we can design a wrapper class to implement a list:

```
public class MyList {

    private int size;
    private Node head;
    ...
}
```

```
MyList numbers = new MyList();
numbers.addAtStart(14);
numbers.addAtStart(6);
numbers.addAtStart(22);
```



Questions (15 min) Start time:

1. In `MyList`, how many assignment operations are required to add 14 *at the front* of an empty list? Note that creating a `Node` takes two assignments (one for `value` and one for `next`).

2. In `MyList`, how many operations are required to add 22 at the front, after 14 and 6 have been added?

3. How many operations are required to add an element *at the end* of `MyList` with 3 elements?

4. How much memory is needed to store each element in the `LinkedList`? How does that amount compare with using an `ArrayList`?

5. Discuss why `LinkedList` is a poor choice of `List` in the program below.

```
1  import java.util.LinkedList;
2  import java.util.List;
3
4  public class LinksAreBad {
5
6      public static void main(String[] args) {
7          List<String> list = new LinkedList<>();
8          System.out.println("Start");
9          addAndGet(list);
10         System.out.println("Done!");
11     }
12
13     public static void addAndGet(List<String> list) {
14         for (int i = 0; i < 1000000; i++) {
15             list.add("A");  // add at the end
16         }
17         for (int i = 0; i < 1000000; i++) {
18             list.get(list.size() / 2);  // get the middle
19         }
20     }
21 }
```

6. If your program requires a `List` collection, how would you decide which implementation to use? (`ArrayList` vs `LinkedList`)