

### Linked list

- A linked list is a data structure that stores a sequence of objects and can quickly add or remove one object.
- Rather than using an array, a linked list uses nodes to store a value and has a reference determined by the nodes next to it. Whereas adding or removing an element alters the reference to every element that follows it, adding or removing a node only affects the neighboring elements.
- Similar to an ArrayList, the LinkedList class is a generic class, meaning it can store whatever type object is specified by the programmer.
- The LinkedList javadoc can be found here:

<http://download.oracle.com/javase/6/docs/api/java/util/LinkedList.html>

- To traverse a linked list, a new ListIterator<Object> must be constructed and then called. For example, to sort through LinkedList<String> strings, you must write:

```
ListIterator<String> iter = strings.listIterator();  
while(iter.hasNext()) iter.next();
```

It is smart to check if the list has another node before calling the next() method.

### Implementing Linked Lists

- A node stores both an object (node.data) and a reference to the object next to it (node.next). These instance variables are kept public because they are accessed frequently.
- The processes of adding a removing are very similar in code. For example, the addFirst method:

```
public void addFirst(Object element) {  
    //creates the new Node.  
    Node newNode = new Node();  
    //sets the object of the node.  
    newNode.data = element;  
    //moves the old first node next to the new node, and sets the reference to the new nodes neighbor  
    newNode.next = first;  
    //places the new node into the first position  
    first = newNode;  
}
```

Other adding and removing methods are similar, editing the “location” of the new element’s neighbor and then placing the new node in the correct place.

- The LinkedList class uses the listIterator() method to create a new LinkedListIterator. This implements the interface ListIterator, which contains the methods used by the LinkedListIterator.

### Abstract Data Types:

Viewing the abstract data types helps indicate the efficiency of their implementations. An ArrayList implements both the abstract interfaces Array and List. These efficiencies are as follows:

Operation	Abstract Array	Abstract List
Random Access	O(1)	O(n)
Linear Traversal Step	O(1)	O(1)
Add/Remove an element	O(n)	O(1)