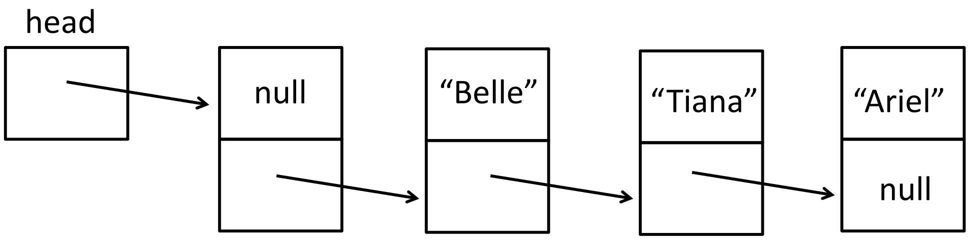
```
public class GenericList<E>
{

     private static class Node<E>
     {
          public E data;
          public Node<E> next;

          public Node(E data, Node<E> next)
          {
               this.data = data;
               this.next = next;
          }
     }

     // Points to the front of the list.
     private Node<E> head;

     // initialize an empty list
     public GenericList()
     {
          head = new Node<E>( null, null );
     }


     // addToFront() adds an element to the front of this list.
     public void addToFront(E e)
     {
          Node<E> newNode = new Node<E>( e, null );

          -----------------------------------------

          -----------------------------------------
     }


     public void mysteryMethod(E e)
     {
          Node<E> current = head;

          while ( current.next != null )
               current = current.next;

          Node<E> newNode = new Node<E>( e, null );
          current.next = newNode;
     }
}
```

**ICS/CSE 22** **Monday, April 11, 2011**

# Quiz 2

*Instructor: Sandy Irani*

Suppose we use the class definition on the opposite page to create an instance of `GenericList<String>`. The list stores a sequence of strings in a singly linked list. It has a dummy node at the front of the list which does not store valid data as we have discussed in class. The **front** of the list is to the left and the **back** of the list is to the right. Suppose we have a singly linked list which stores the sequence "Belle", "Tiana", "Ariel". It would look like:



1. The method `addToFront()` adds a new item to the front of the list. Fill in the two missing lines on the opposite page to make the method work.

2. Show what the list looks like after we call `mysteryMethod("Mulan");`.

3. Describe in words what `myseteryMethod` does. Be specific.