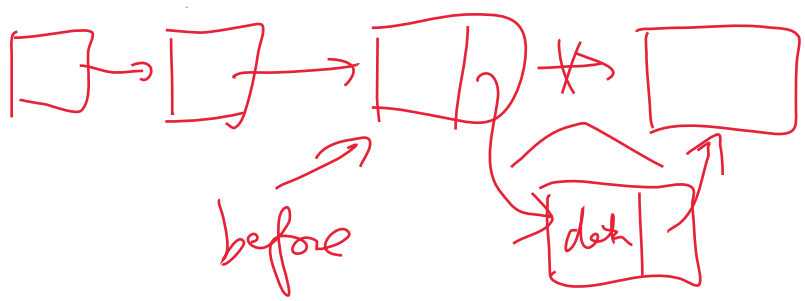
Iterator worksheet

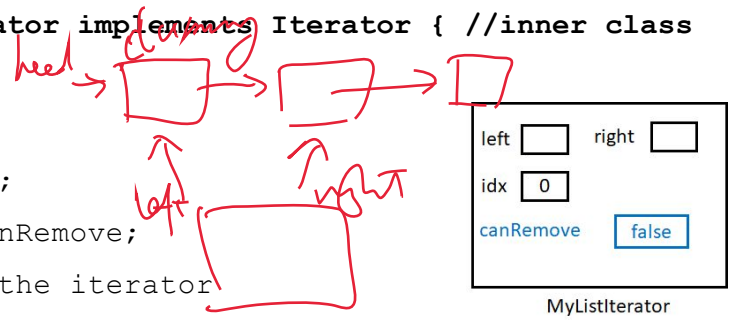
Iterator is an interface that has some method to retrieve elements from a collection object one by one. Let's use the Iterator interface to create a list of friends such that it iterates over the friends that begin with 'a'.

.

```
//worksheet for linkedlist and iterator
import java.io.*;
import java.util.*;
interface CSE12List<E>{
  public void insert(int index, E element);
  public void print();
  //other functions such as remove, find, etc
}
class FriendList<E> implements CSE12List<E>{
  //Inner class that is a node
  class Node{ //inner class
    E data;
    Node next;
    public Node(){
      data = null;
      next = null;
    }
    public Node(E data, Node before){
      if(before != null){
        this.data = data; //assign data
        this.next = before.next; //link to the element behind
        before.next = this; //link from before
      }
    }
    public Node Next(){
      return next;
    }
  }
```



```
class FriendListIterator implements Iterator { //inner class
  private int index;
  private Node left;
  private Node right;
  private boolean canRemove;
  //constructor for the iterator
  public FriendListIterator() {
    left = head                          //initialize left
    right = head.Next()                  //initialize right
    index = 0                            //iniitalize index
    canRemove = false                    //initialize canRemove
  }
  //override next method
  public boolean hasNext() {
    return index < size                  //decision based on size
  }
  //next method
  public E next() {
    E result = null;
    if (size == 0){
        return null;
    }
    if (((String)right.data).startsWith("a")){//for execise only
        result = right.data;
    }
    if (right.next != null){ //move to the next element
        left = left.Next()
        right = right.Next()
        index ++;
        canRemove = true
    }
    return result;
  }
}
```



```
68.     //instance variables for FriendList
69.     private Node head;
70.     private Node tail;
71.     private int size;
72.
73.     //iterator method to get an interator
74.     public Iterator<E> iterator() {
75.       return ____new FriendList Iterator ();____
76.     }
77.     //other methods for FriendList
78.   }
```