

```

public class BasicLinkedList<X> {
    private Node first;
    private Node last;
    private int nodeCount ;

    public BasicLinkedList(){
        first = null;
        last = null;
        nodeCount = 0;
    }

    public void add(X item){

        if(first == null){
            first = new Node(item);
            last = first;
        }
        else{
            //create the Node
            Node newLastNode = new Node(item);
            //add it to the current last Node
            last.setNextNode(newLastNode);
            //mark current node as the last Node
            last = newLastNode;
        }

        nodeCount++;
    }

    public X remove(){
        if(first == null){
            throw new IllegalStateException("list is empty and
there are no new items to remove");
        }
        else{
            // get the first Node
            X nodeItem = first.getNodeItem();
            // set the next Node as the first node

```

```

        first = first.getNextNode();
        nodeCount--;
        return nodeItem;
    }

}

public int size(){
    return nodeCount;
}

private class Node{
    private Node nextNode;
    private X nodeItem;

    public Node(X item){
        this.nextNode = null;
        this.nodeItem = item;
    }

    public void setNextNode(Node nextNode){
        this.nextNode = nextNode;
    }

    public Node getNextNode(){
        return nextNode;
    }

    public X getNodeItem(){
        return nodeItem;
    }
}
}

```