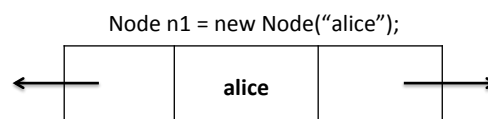
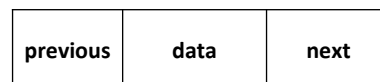


Doubly Linked List

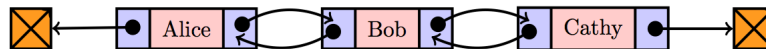
Doubly Linked List Node

```
class Node{  
    E data;  
    Node previous;  
    Node next;  
    Node(E item){  
        data = item;  
        next = null;  
        previous = null;  
    }  
    Node(){  
        data = null;  
        next = null;  
        previous = null;  
    }  
}
```

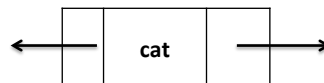
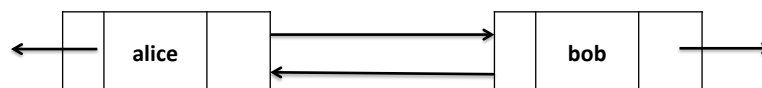


Doubly Linked List

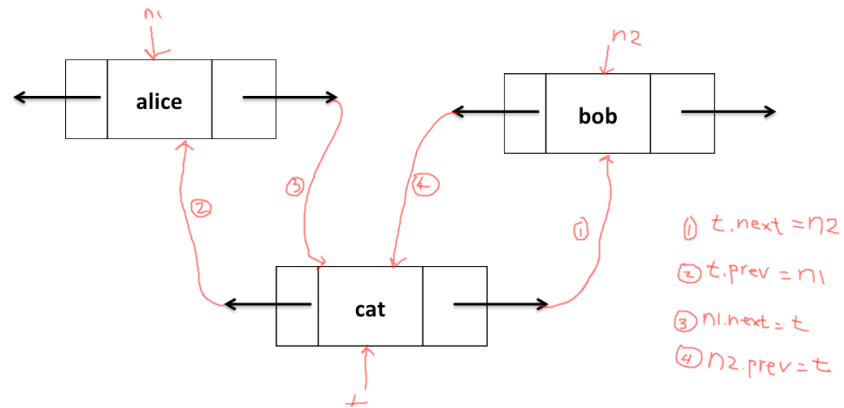
```
Node<String> n1 = new Node(" Alice" );
Node<String> n2 = new Node(" Bob" );
Node<String> n3 = new Node(" Cathy" );
n1.next = n2;
n2.previous = n1;
n2.next = n3;
n3.previous = n2;
```



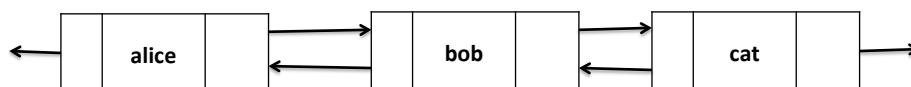
Insert a Node



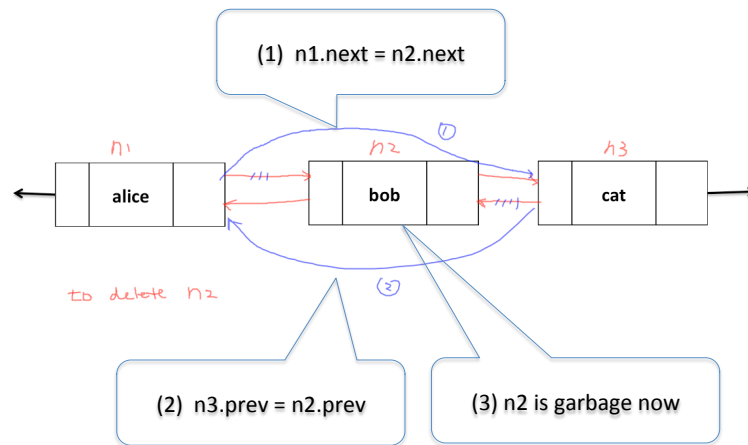
Insert a Node



Delete a Node



Delete a node



Double Linked List Class

```
public class DoublyLinkedList<E> implements Iterable<E>{
    private int N; // number of nodes
    private Node head; //sentinel before the first node
    private Node tail; //sentinel after the last node;
    DoublyLinkedList(){
        head = new Node();
        tail = new Node();
        head.next = tail;
        tail.previous = head;
    }
}
```

Insert

```
public void insert(E item){  
    Node last = tail.previous;  
    Node t = new Node(item);  
    t.next = tail;  
    t.previous = last;  
    tail.previous = t;  
    last.next = t;  
    N++;  
}
```

