

## Class Student

```
public class Student {
    private int id;
    private String name;
    private double mark;
    public Student(int id, String name, double mark) {
        this.id = id;
        this.name = name;
        this.mark = mark;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getMark() {
        return mark;
    }
    public void setMark(double mark) {
        this.mark = mark;
    }
    public void display(){
        System.out.println("Student id: " + id);
        System.out.println("Student name: " + name);
        System.out.println("Student mark: " + mark);
    }
}
```

## Class Node

```
public class Node {
    private Student data;
    private Node next;
    public Node(){
        data = null;
        next = null;
    }
    public Node(Student data){
        this.data = data;
        this.next = null;
    }
    public Node(Student data, Node next){
        this.data = data;
        this.next = next;
    }
    public Student getData() {
        return data;
    }
    public void setData(Student data) {
        this.data = data;
    }
    public Node getNext() {
        return next;
    }
    public void setNext(Node next) {
        this.next = next;
    }
}
```

## Class LinkedList

```
public class LinkedList {
    private Node head;
    public LinkedList(){
        this.head = null;
    }
    public boolean isEmpty(){
        return head == null;
    }
    public void insertAtFront(Student s){
        Node newNode = new Node(s);
        newNode.setNext(head);
        head = newNode;
    }
    public void insertAtEnd(Student s){
        Node newNode = new Node(s);
        if(isEmpty()){
            head = newNode;
            return;
        }
        Node current = head;
        while(current.getNext() != null)
            current = current.getNext();
        current.setNext(newNode);
    }
    public void removeFromFront(){
        if(!isEmpty())
            head = head.getNext();
    }
    public Student removeFromFront2(){
        Student temp = null;
        if(!isEmpty()){
            temp = head.getData();
            head = head.getNext();
        }
        return temp;
    }
    public void removeFromBack(){
        if(isEmpty()) return;
        if(head.getNext() == null){
            head = null;
            return;
        }
        Node current = head;
        Node previous = null;
        while(current.getNext() != null){
            previous = current;
            current = current.getNext();
        }
    }
}
```

```

        previous.setNext(null);
    }
    public Student removeFromBack2(){
        if(isEmpty()) return null;
        if(head.getNext() == null){
            Student temp = head.getData();
            head = null;
            return temp;
        }
        Node current = head;
        Node previous = null;
        while(current.getNext() != null){
            previous = current;
            current = current.getNext();
        }
        Student temp = current.getData();
        previous.setNext(null);
        return temp;
    }
    public int size(){
        int size = 0;
        Node current = head;
        while(current != null){
            size++;
            current = current.getNext();
        }
        return size;
    }
    public void display(){
        Node current = head;
        while(current != null){
            current.getData().display();
            current = current.getNext();
        }
    }
    public boolean searchById(int id){
        Node current = head;
        while(current != null){
            if(current.getData().getId() == id)
                return true;
            current = current.getNext();
        }
        return false;
    }
}
}

```