LINKED LIST DATA STRUCTURE (LAB 3)

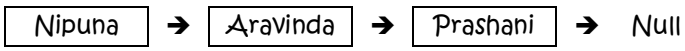
Question a

i. Implement Link class to store the name and average of a student. Identify the data attributes required and implement displayDetails() method with the constructor

ii. In an application, create three links with the following details

Name	Average
Nipuna	53.5
Aravinda	78.0
Prashani	69.5

iii. Connect the three links as follows.

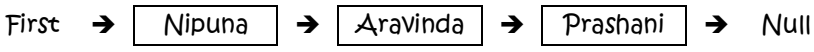


Question b

i. Implement a LinkList class with the following data members and method

LinkList
- first
- LinkList()
- isEmpty()
- displayList()
- insertFirst()
- deleteFirst()

ii. In your application, create the following link using the LinkList class and display the link details.



Hint : Use insertFirst() method

iii. Add more links to your link list.

Question c

i. Modify your linkList class to add a new method called deleteLink(). This method will find the link with the given name and delete the link.

Link deleteLink(String name)

ii. Modify your application to delete a given link and display the details of the list.

Answers:

```
public class LinkList {
	private Link first;

	public LinkList() {
		first = null;
	}//constructor

	public boolean isEmpty() {
		return (first == null);
	}//isEmpty method

	public void displayList() {
		Link current = first;
		while (current != null) {
			current.displayDetails();
			current = current.next;
		}
	}//displayList method
```

```
public class Link {
	public String name;
	public double average;
	public Link next;

	public Link(String n, double a) {
		this.name = n;
		this.average = a;
	}//constructor

	public void displayDetails() {
		System.out.println(name + " " + average);
	}//displayDetails method

}//class
```

```
public void insertFirst(String n, double a) {
		Link l = new Link(n, a);
		l.next = first;
		first = l;
	}//insertFirst method

	public Link deleteFirst() {
		Link temp = first;
		first = first.next;
		return temp;
	}//deleteFirst method

	public Link deleteLink(String n) {
		Link current = first;
		Link previous = first;

		while (current.name != n) {
			if (current.next == null) {	//name does not exist in the list
				return null;
			} else {
				previous = current;
				current = current.next;
				previous.next = current.next;
			}//if-else
		}//while - traverse through the list till the name is found

		if (current == first) {
			deleteFirst();
		}//if  name is found in the fisrt position

		return current;
	}//deleteLink

}//class
```

```
public class Application {

	public static void main(String[] args) {
		Link l1 = new Link("Nipuna", 53.5);
		Link l2 = new Link("Aravinda", 78.0);
		Link l3 = new Link("Prashani", 69.5);  //question a ii

		l1.next = l2;
		l2.next = l3;
		l3.next = null;			//question a iii

		l1.displayDetails();
		l2.displayDetails();
		l3.displayDetails();	//test display results
		System.out.println("");

		LinkList list = new LinkList();

		list.insertFirst("Amal", 89.2);
		list.insertFirst("Nimal", 44.2);
		list.insertFirst("Prashani", 69.5);
		list.insertFirst("Aravinda", 78.0);
		list.insertFirst("Nipuna", 53.5);

		list.displayList();
		System.out.println("");
		list.deleteFirst();
		list.displayList();
		System.out.println("");
		list.deleteLink("Prashani");	//try with Aravinda
		list.displayList();
	}//main
}//class
```