Linked List practical

Using Default Java Class

```
// Java code for Linked List implementation

import java.util.*;

public class Test
{
	public static void main(String args[])
	{
		// Creating object of class linked list
		LinkedList<String> object = new LinkedList<String>();

		// Adding elements to the linked list
		object.add("A");
		object.add("B");
		object.addLast("C");
		object.addFirst("D");
		object.add(2, "E");
		object.add("F");
		object.add("G");
		System.out.println("Linked list : " + object);

		// Removing elements from the linked list
		object.remove("B");
		object.remove(3);
		object.removeFirst();
		object.removeLast();
		System.out.println("Linked list after deletion: " + object);

		// Finding elements in the linked list
		boolean status = object.contains("E");
```

```
            if(status)
                System.out.println("List contains the element 'E' ");
            else
                System.out.println("List doesn't contain the element 'E'");

            // Number of elements in the linked list
            int size = object.size();
            System.out.println("Size of linked list = " + size);

            // Get and set elements from linked list
            Object element = object.get(2);

            System.out.println("Element returned by get() : " + element);
            object.set(2, "Y");
            System.out.println("Linked list after change : " + object);
        }
}
```

Simple Linked List

```
class
LinkedList
                {
                    //Class variables for the Linked List
```

```
        private static Node head;
        private static int numNodes;

        public static void main(String [] args)
        {
                System.out.println("/=/=/=/= TESTING
/=/=/=/=");
                LinkedList ll = new LinkedList(10);
                ll.addAtHead(11);
                ll.addAtHead(12);
                ll.addAtHead(13);
                ll.addAtTail(8);
                ll.addAtTail(7);
                ll.addAtIndex(4,9);
                ll.addAtIndex(4,9);
                ll.deleteAtIndex(4);
                ll.printList();
        }

        public LinkedList(Object dat)
        {
                head = new Node(dat);
        }
```

```
public void addAtHead(Object dat)
{
        Node temp = head;
        head = new Node(dat);
        head.next = temp;
        numNodes++;
}

public void addAtTail(Object dat)
{
        Node temp = head;
        while(temp.next != null)
        {
                temp = temp.next;
        }

        temp.next = new Node(dat);
        numNodes++;
}

public void addAtIndex(int index, Object dat)
{
        Node temp = head;
        Node holder;
```

```
            for(int i=0; i < index-1 && temp.next !=
null; i++)
            {
                temp = temp.next;
            }
            holder = temp.next;
            temp.next = new Node(dat);
            temp.next.next = holder;
            numNodes++;
        }

        public void deleteAtIndex(int index)
        {
            Node temp = head;
            for(int i=0; i< index - 1 && temp.next !=
null; i++)
            {
                temp = temp.next;
            }
            temp.next = temp.next.next;
            numNodes--;
        }

        public static int find(Node n)
```

```
{
        Node t = head;
        int index = 0;
        while(t != n)
        {
                index++;
                t = t.next;
        }
        return index;
}

public static Node find(int index)
{
        Node temp=head;
        for(int i=0; i<index; i++)
        {
                temp = temp.next;
        }
        return temp;
}

public static void printList()
{
        Node temp = head;
```

```
        while(temp != null)
        {
                System.out.println(temp.data);
                temp = temp.next;
        }
}

public static int getSize()
{
        return numNodes;
}

class Node
{
        //Declare class variables
        private Node next;
        private Object data;

        public Node(Object dat)
        {
                data = dat;
        }

        public Object getData()
```

```
                {
                    return data;
                }
            }
        }
```