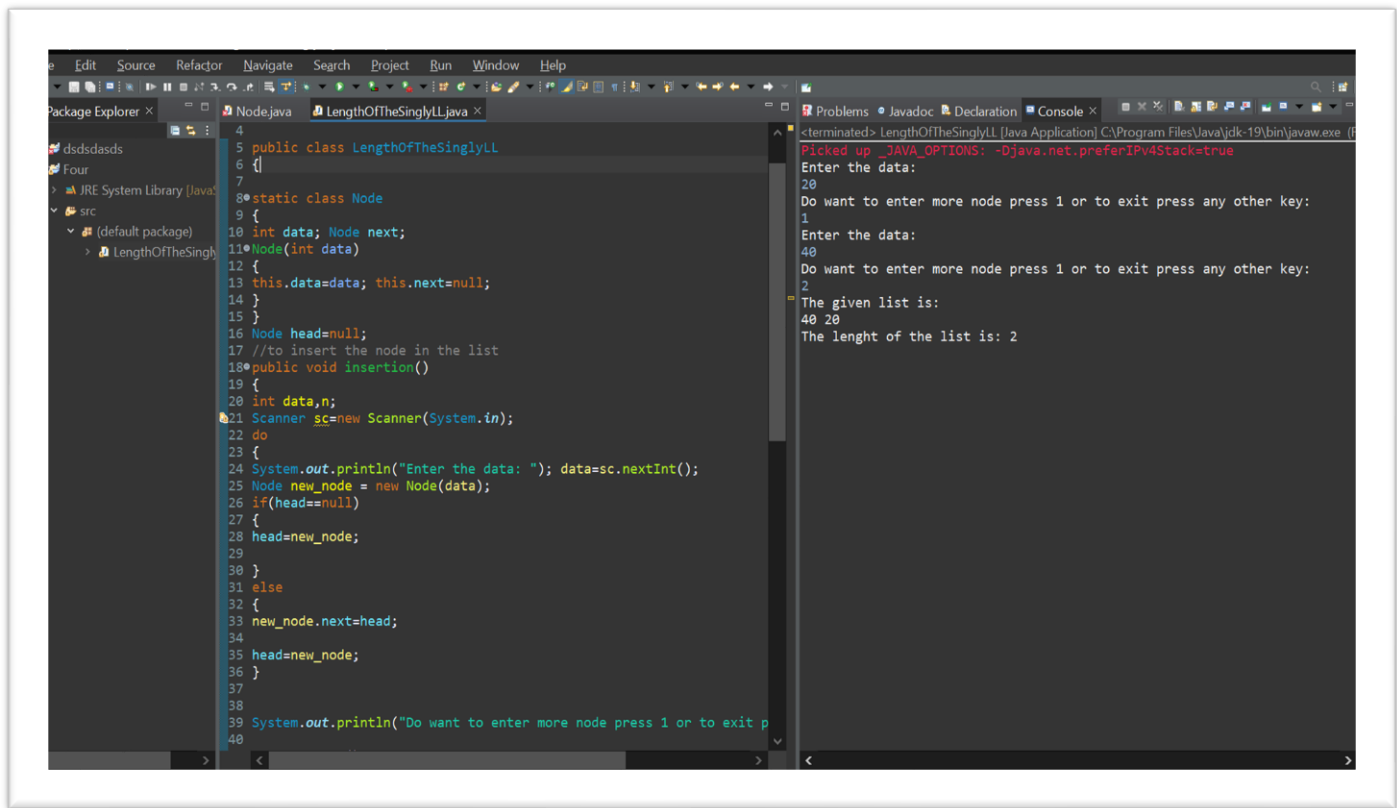


## LAB SHEET -4

### AIM 1: Understanding the concepts Linked List (10points)

1. Implement a program to count the length of a singly linked list.



```
4 public class LengthOfTheSinglyLL
5 {
6 }
7
8 static class Node
9 {
10 int data; Node next;
11 Node(int data)
12 {
13 this.data=data; this.next=null;
14 }
15 }
16 Node head=null;
17 //to insert the node in the list
18 public void insertion()
19 {
20 int data,n;
21 Scanner sc=new Scanner(System.in);
22 do
23 {
24 System.out.println("Enter the data: "); data=sc.nextInt();
25 Node new_node = new Node(data);
26 if(head==null)
27 {
28 head=new_node;
29 }
30 }
31 else
32 {
33 new_node.next=head;
34 }
35 head=new_node;
36 }
37 }
38
39 System.out.println("Do want to enter more node press 1 or to exit p
40
```

Console Output:

```
<terminated> LengthOfTheSinglyLL [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (t
Picked up _JAVA_OPTIONS: -Djava.net.preferIPv4Stack=true
Enter the data:
20
Do want to enter more node press 1 or to exit press any other key:
1
Enter the data:
40
Do want to enter more node press 1 or to exit press any other key:
2
The given list is:
40 20
The lenght of the list is: 2
```

Code:

```
import java.util.Scanner;

public class LengthOfTheSinglyLL
{
    static class Node
    {
        int data; Node next;
```

```
Node(int data)
{
    this.data=data; this.next=null;
}

Node head=null;

//to insert the node in the list

public void insertion()
{
    int data,n;

    Scanner sc=new Scanner(System.in);

    do
    {
        System.out.println("Enter the data: "); data=sc.nextInt();

        Node new_node = new Node(data);

        if(head==null)
        {
            head=new_node;
        }

        else
        {
            new_node.next=head;

            head=new_node;
        }

        System.out.println("Do want to enter more node press 1 or to exit press any other key: ");
    }
```

```
n=sc.nextInt();

}

while(n==1);

}

//to display the elemements available in the list

public void display()

{

Node temp=head; System.out.println("The given list is: "); while(temp!=null)

{

System.out.print(temp.data + " "); temp=temp.next;

}

System.out.println();

}

//To get lenght

public int getCount()

{

Node temp = head;

int count = 0;

while (temp != null)

{

count++;

temp = temp.next;

}

return count;

}
```

```

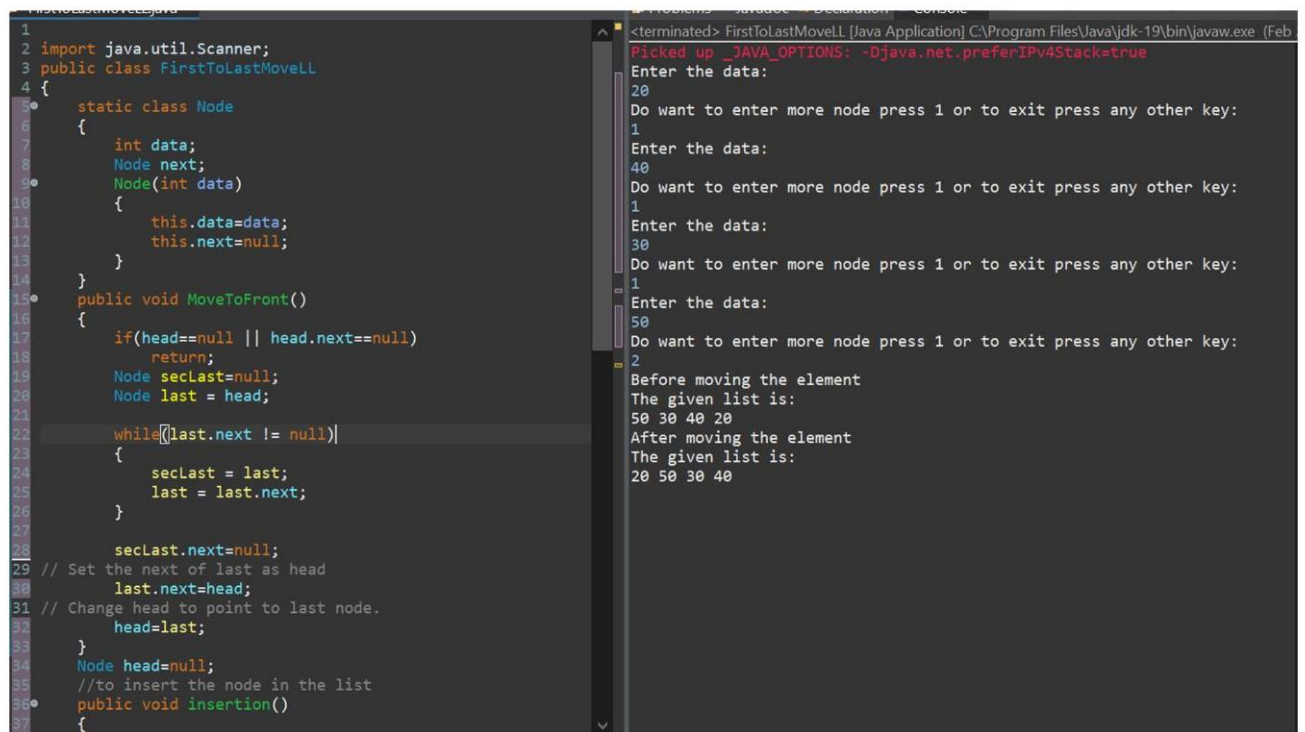
public static void main(String[] args)
{
    LengthOfTheSinglyLL ll = new LengthOfTheSinglyLL(); ll.insertion();

    ll.display();

    System.out.println("The length of the list is: "+ll.getCount());
}
}

```

## 2. Implement a program to move last node of a singly linked list to front.



```

1  import java.util.Scanner;
2  public class FirstToLastMoveLL
3  {
4      static class Node
5      {
6          int data;
7          Node next;
8          Node(int data)
9          {
10             this.data=data;
11             this.next=null;
12         }
13     }
14     public void MoveToFront()
15     {
16         if(head==null || head.next==null)
17             return;
18         Node secLast=null;
19         Node last = head;
20         while(last.next != null)
21         {
22             secLast = last;
23             last = last.next;
24         }
25         secLast.next=null;
26         // Set the next of last as head
27         last.next=head;
28         // Change head to point to last node.
29         head=last;
30     }
31     Node head=null;
32     //to insert the node in the list
33     public void insertion()
34     {
35
36     }
37 }

```

<terminated> FirstToLastMoveLL (Java Application) C:\Program Files\Java\jdk-19\bin\javaw.exe (Feb  
 Picked up \_JAVA\_OPTIONS: -Djava.net.preferIPv4Stack=true  
 Enter the data:  
 20  
 Do want to enter more node press 1 or to exit press any other key:  
 1  
 Enter the data:  
 40  
 Do want to enter more node press 1 or to exit press any other key:  
 1  
 Enter the data:  
 30  
 Do want to enter more node press 1 or to exit press any other key:  
 1  
 Enter the data:  
 50  
 Do want to enter more node press 1 or to exit press any other key:  
 2  
 Before moving the element  
 The given list is:  
 50 30 40 20  
 After moving the element  
 The given list is:  
 20 50 30 40

### Code:

```

import java.util.Scanner;

public class FirstToLastMoveLL

```

```
{  
  
static class Node  
  
{  
  
int data;  
  
Node next;  
  
Node(int data)  
  
{  
  
this.data=data;  
  
this.next=null;  
  
}  
  
}  
  
public void MoveToFront()  
  
{  
  
if(head==null || head.next==null)  
  
return;  
  
Node secLast=null;  
  
Node last = head;  
  
while(last.next != null)  
  
{  
  
secLast = last;  
  
last = last.next;  
  
}  
  
secLast.next=null;  
  
// Set the next of last as head  
  
last.next=head;  
  
}
```

```
// Change head to point to last node.

head=last;

}

Node head=null;

//to insert the node in the list

public void insertion()

{

    int data,n;

    Scanner sc=new Scanner(System.in);

    do

    {

        System.out.println("Enter the data: ");

        data=sc.nextInt();

        Node new_node = new Node(data);

        if(head==null)

        {

            head=new_node;

        }

        else

        {

            new_node.next=head;

            head=new_node;

        }

        System.out.println("Do want to enter more node press 1 or to exit press any other key: ");

        n=sc.nextInt();

    }
```

```
}  
  
while(n==1);  
  
}  
  
public void display()  
{  
    Node temp=head;  
  
    System.out.println("The given list is: ");  
  
    while(temp!=null)  
    {  
        System.out.print(temp.data + " ");  
        temp=temp.next;  
    }  
  
    System.out.println();  
}  
  
public static void main(String[] args)  
{  
    FirstToLastMoveLL link1=new FirstToLastMoveLL();  
    link1.insertion();  
  
    System.out.println("Before moving the element");  
  
    link1.display();  
  
    link1.MoveToFront();  
  
    System.out.println("After moving the element");  
  
    link1.display();  
}  
}
```

