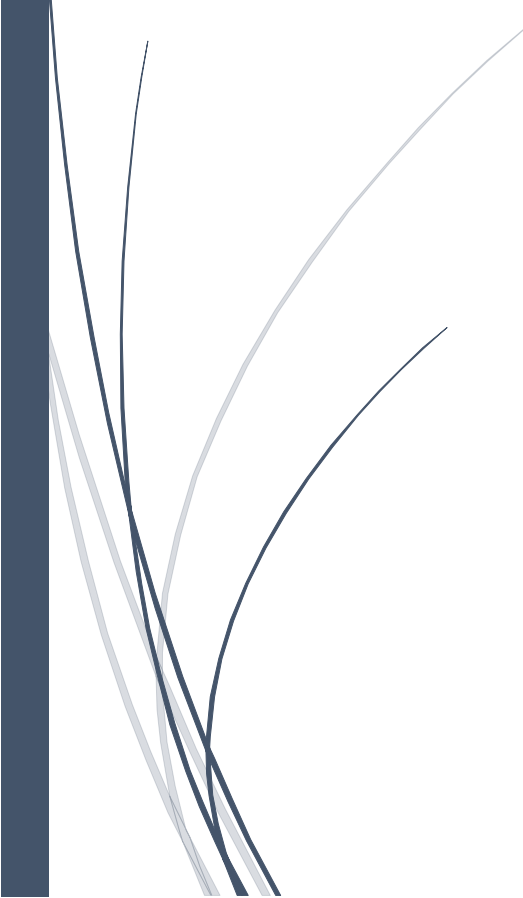


A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

9/1/2024

# Data Structures

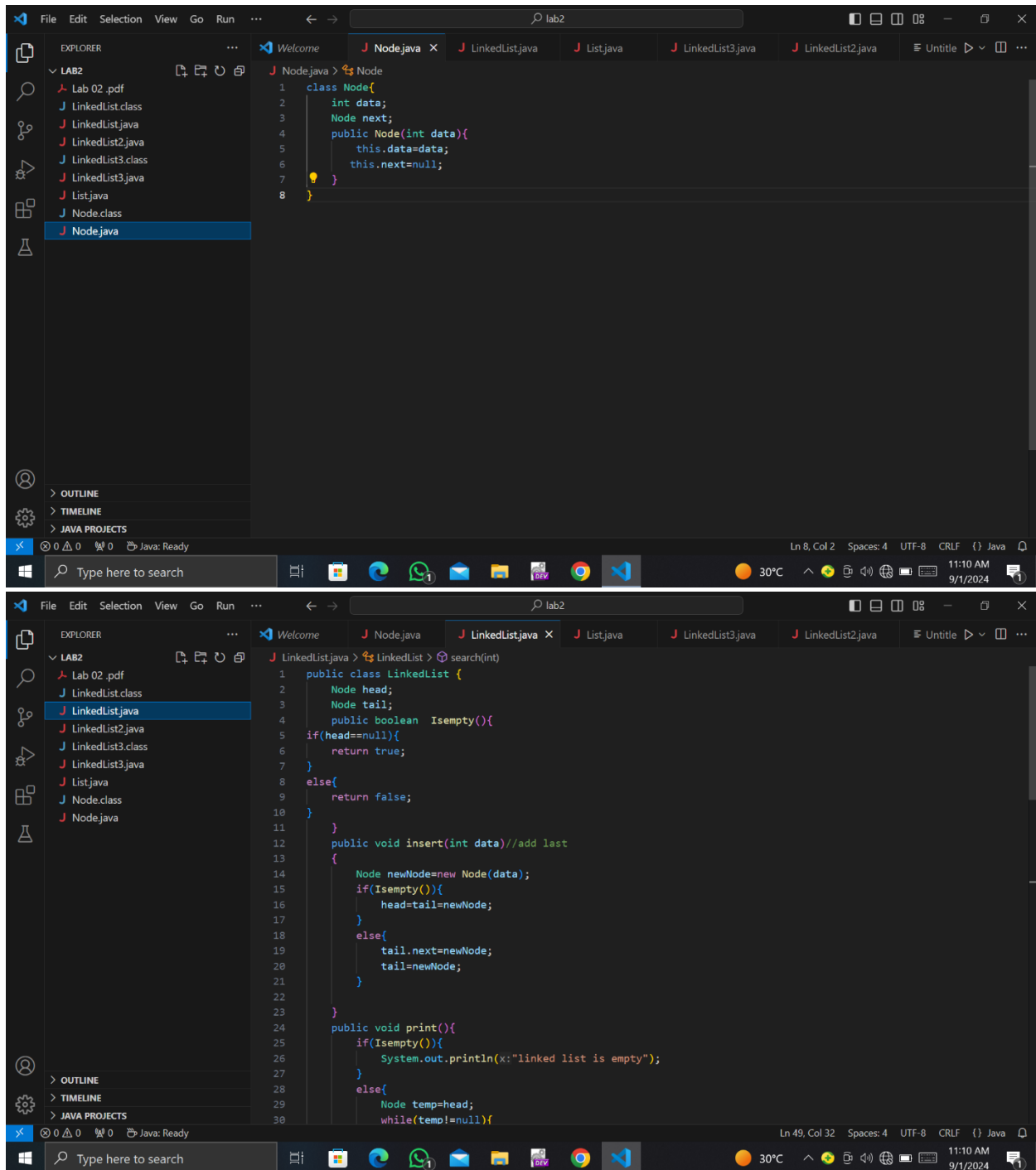
## Lab 2 Exercise Solutions

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

Uzair Ali S/O Yar Muhammad Memon  
B-S CS III Section A  
023-23-0350  
To: Ma'm Marina Rajpoot

## Task #1

## Solution:



```
31         System.out.print(temp.data+" ");
32         temp=temp.next;
33     }
34     System.out.println();
35 }
36
37 public void search(int element){
38     if(isempty()){
39         System.out.println(x:"linked list is empty");
40     }
41     else{
42         Node temp=head;
43         int index=1;
44         while(temp!=null){
45             if(element==temp.data){
46                 System.out.println("element "+element+" is found at index : "+index);
47                 break;
48             }
49             temp=temp.next;
50             index++;
51             if(temp==null){
52                 System.out.println("element "+element+" is not found");
53             }
54         }
55     }
56 }
57
58
59 }
```

```
60 public static void main(String[] args) {
61     LinkedList ll =new LinkedList();
62     ll.print();
63     ll.insert(data:5);
64     ll.insert(data:6);
65     ll.insert(data:7);
66     ll.insert(data:10);
67     ll.print();
68     ll.search(element:6);
69 }
```

Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.

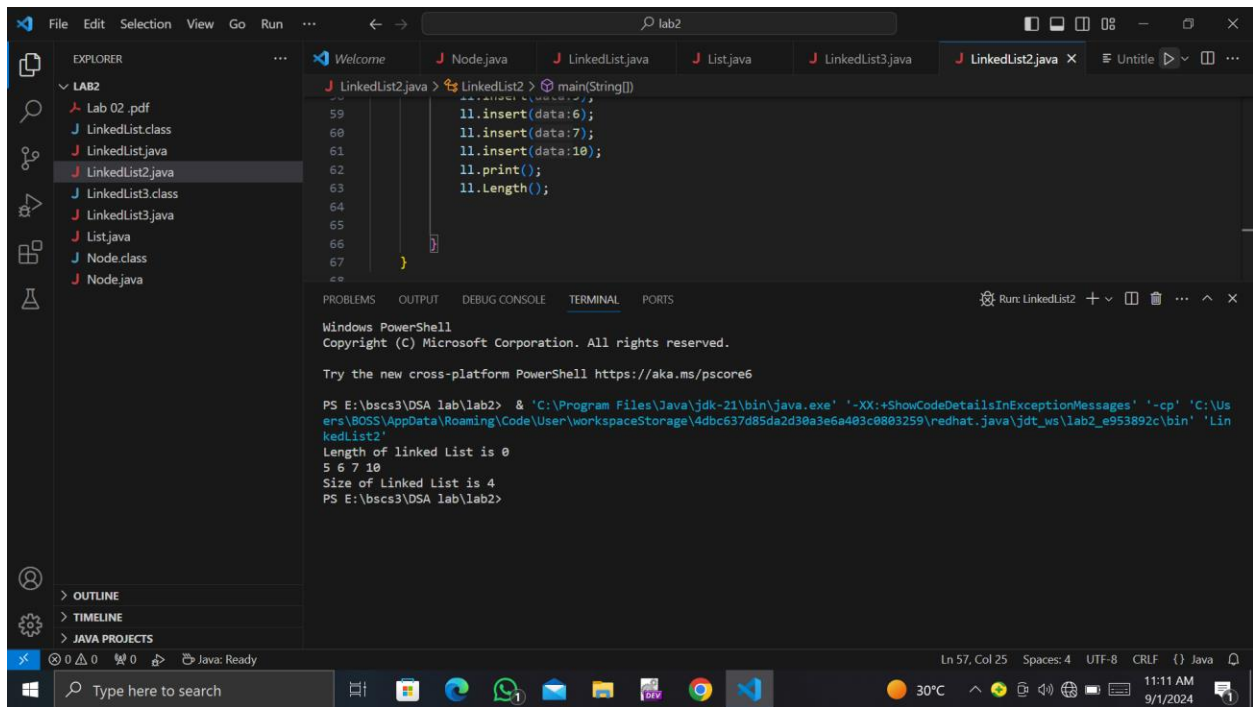
Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\lab2> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\4dbc637d85da2d38a3e6a403c0803259\redhat.java\jdt_ws\lab2_e953892c\bin' 'LinkedList'
linked list is empty
5 6 7 10
element 6 is found at index : 2
PS E:\bscs3\DSA lab\lab2>
```

## Task #2

### Solution:

```
1 public class LinkedList2 {
2     Node head;
3     Node tail;
4     public boolean isEmpty(){
5         if(head==null){
6             return true;
7         }
8         else{
9             return false;
10        }
11    }
12    public void insert(int data)//add last
13    {
14        Node newNode=new Node(data);
15        if(isEmpty()){
16            head=tail=newNode;
17        }
18        else{
19            tail.next=newNode;
20            tail=newNode;
21        }
22    }
23    }
24    public void print(){
25        if(isEmpty()){
26            System.out.println(x:"linked list is empty");
27        }
28        else{
29            Node temp=head;
30            while(temp!=null){
31                System.out.print(temp.data+" ");
32                temp=temp.next;
33            }
34            System.out.println();
35        }
36    }
37    public void Length(){
38        int size=0;
39        if(isEmpty()){
40            System.out.println("Length of linked List is "+size);
41        }
42        else{
43            Node temp=head;
44            while(temp!=null){
45                size++;
46                temp=temp.next;
47                if(temp==null){
48                    System.out.println("Size of Linked List is "+ size);
49                }
50            }
51        }
52    }
53    }
54    }
55    Run | Debug
56    public static void main(String[] args) {
57        LinkedList2 l1 =new LinkedList2();
58        l1.Length();
59        l1.insert(data:5);
60        l1.insert(data:6);
61    }
```



## Task #3

## Solution:

```
1  
2  
3 public class LinkedList3 {  
4     Node head;  
5     int size;  
6     public LinkedList3(){  
7         head = new Node(data:0);  
8         size = 0;  
9     }  
10    public boolean isEmpty(){  
11        if (head.data==0){  
12            return true;  
13        }  
14        return false;  
15    }  
16    public int size(){  
17        return size;  
18    }  
19    public void add(int data){  
20        Node newnode=new Node(data);  
21        if(isEmpty()){  
22            head=newnode;  
23            size++;  
24        }else{  
25            Node temp=head;  
26            while(temp.next!=null){  
27                temp=temp.next;  
28            }  
29            temp.next=newnode;  
30            size++;  
31        }  
32    }  
33    public void add(int index, int data){  
34        Node newnode=new Node(data);  
35        if(index==1){  
36            newnode.next=head;  
37            head=newnode;  
38            size++;  
39        }  
40        else{  
41            newnode.data=data;  
42            Node temp=head;  
43            for(int i=1;i<index-1;i++){  
44                temp=temp.next;  
45            }  
46            newnode.next=temp.next;  
47            temp.next=newnode;  
48            size++;  
49        }  
50    }  
51    public void removefromIndex(int index){  
52        if (index==1){  
53            head=head.next;  
54        }  
55        else{  
56            Node temp=head;  
57            for(int i=1;i<index-1;i++){  
58                temp=temp.next;  
59            }  
60            temp.next=temp.next.next;
```

```
61         size--;
62     }
63 }
64 void remove(int item) {
65     if (head == null) {
66         return;
67     }
68
69     // If the item is at the head
70     if (head.data==item) {
71         head = head.next;
72         return;
73     }
74
75     Node temp = head;
76     while (temp.next != null) {
77         if (temp.next.data==item) {
78             // If the node to remove is the last one
79             if (temp.next.next == null) {
80                 temp.next = null;
81                 size--;
82             } else {
83                 temp.next = temp.next.next;
84                 size--;
85             }
86             return;
87         }
88         temp=temp.next;
89     }
90     LinkedList3 duplicate(){
```

```
91     LinkedList3 duplicate=new LinkedList3();
92     Node temp=head;
93     while(temp!=null){
94         duplicate.add(temp.data);
95         temp=temp.next;
96     }
97     return duplicate;
98 }
99 public void print(){
100     if(isEmpty()){
101         System.out.println(x:"linked list is empty");
102     }
103     else{
104         Node temp=head;
105         while(temp!=null){
106             System.out.print(temp.data+" ");
107             temp=temp.next;
108         }
109         System.out.println();
110     }
111 }
112 public void AddFront(int data){
113     Node newNode=new Node(data);
114     if(isEmpty()){
115         head=newNode;
116     }
117     else{
118         newNode.next=head;
119         head=newNode;
120     }
```

The screenshot shows the VS Code editor with the file `LinkedList3.java` open. The code defines a `LinkedList3` class with a `ReversedList()` method that reverses the linked list. The `main` method tests the list operations.

```
120 }
121 }
122 }
123 LinkedList3 ReversedList(){
124     LinkedList3 reversedlist=new LinkedList3();
125     Node temp=head;
126     while(temp!=null){
127         reversedlist.AddFront(temp.data);
128         temp=temp.next;
129     }
130     return reversedlist;
131 }
132 }
Run | Debug
133 public static void main(String[] args) {
134     LinkedList3 ll=new LinkedList3();
135     ll.add(data:3);
136     ll.add(data:4);
137     ll.add(data:5);
138     ll.add(data:8);
139     ll.add(index:3, data:10);
140     ll.print();
141     System.out.println(ll.size());
142     ll.removefromIndex(index:3);
143     ll.print();
144     System.out.println(ll.size());
145     ll.remove(item:8);
146     ll.print();
147     System.out.println(ll.size());
148     ll.remove(item:5);
149 }
```

The screenshot shows the VS Code editor with the file `LinkedList3.java` open. The code defines a `LinkedList3` class with a `ReversedList()` method that reverses the linked list. The `main` method tests the list operations. The terminal output shows the execution of the program, displaying the list size and the reversed list.

```
149 ll.print();
150 System.out.println(ll.size());
151 LinkedList3 l=ll.duplicate();
152 l.print();
153 LinkedList3 reverse=ll.ReversedList();
154 reverse.print();
155 }
156 }
157 }
```

Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\lab2> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\4dbc637d85da2d30a3e6a403c0803259\redhat.java\jdt_ws\lab2_e953892c\bin' 'LinkedList3'
3 4 10 5 8
5
3 4 5 8
4
3 4 5
3
3 4
2
3 4
4 3
PS E:\bscs3\DSA lab\lab2>
```

**THE END**