# Data Structures & Algorithms (CSE2001) Lab-3

**KHAN MOHD OWAIS RAZA (20BCD7138)**

Q.1] Write a program to implement single linked list and its operations

```
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
/* Lab-3 (17-09-2022)*/
/* Java code to implement single linked list and its operations */
package CSE2001_Lab3_20BCD7138;
import java.util.Scanner;
class Node{
protected int data;
protected Node link;
public Node(){
link = null;
data = 0;
}
public Node(int d,Node n){
data = d;
link = n;
}
public void setLink(Node n){
link = n;
}
public void setData(int d){
data = d;
}
public Node getLink(){
return link;
}
public int getData(){
return data;
}}
class linkedList{
protected Node start;
protected Node end ;
public int size ;
public linkedList(){
start = null;
end = null;
size = 0;
}
public boolean isEmpty(){
return start == null;
}
public int getSize(){
return size;
```

```
}
public void insertAtStart(int val){
Node nptr = new Node(val, null);
size++ ;
if(start == null){
start = nptr;
end = start;
}
else {
nptr.setLink(start);
start = nptr;
}}
public void insertAtEnd(int val){
Node nptr = new Node(val,null);
size++ ;
if(start == null){
start = nptr;
end = start;
}
else{
end.setLink(nptr);
end = nptr;
}}
public void insertAtPos(int val , int pos){
Node nptr = new Node(val, null);
Node ptr = start;
pos = pos - 1 ;
for (int i = 1; i < size; i++){
if (i == pos){
Node tmp = ptr.getLink() ;
ptr.setLink(nptr);
nptr.setLink(tmp);
break;
}
ptr = ptr.getLink();
}
size++ ;
}
public void deleteAtPos(int pos){
if (pos == 1){
start = start.getLink();
size--;
return ;
}
if (pos == size){
Node s = start;
Node t = start;
while (s != end){
t = s;
s = s.getLink();
}
end = t;
end.setLink(null);
size --;
```

```
return;
}
Node ptr = start;
pos = pos - 1 ;
for (int i = 1; i < size - 1; i++){
if (i == pos){
Node tmp = ptr.getLink();
tmp = tmp.getLink();
ptr.setLink(tmp);
break;
}
ptr = ptr.getLink();
}
size-- ;
}
public void display(){
System.out.print("\nSingle Linked List: ");
if (size == 0){
System.out.print("Empty\n");
return;
}
if (start.getLink() == null){
System.out.println(start.getData() );
return;
}
Node ptr = start;
System.out.print(start.getData()+ " --> ");
ptr = start.getLink();
while (ptr.getLink() != null){
System.out.print(ptr.getData()+ " --> ");
ptr = ptr.getLink();
}
System.out.print(ptr.getData()+ "\n");
}}
public class Question1{
public static void main(String[] args){
Scanner scan = new Scanner(System.in);
linkedList list = new linkedList();
System.out.println("KHAN MOHD OWAIS RAZA (20BCD7138)");
System.out.println("Lab-3 (17-09-2022)");
System.out.println("Java code to implement single linked list and its
operations");
char ch;
do{
System.out.println("\nPlease select the operation:\n");
System.out.println("[1] Insert element at beginning");
System.out.println("[2] Insert element at ending");
System.out.println("[3] Insert element at a position");
System.out.println("[4] Delete element at a position");
System.out.println("[5] Check if the list is empty");
System.out.println("[6] Display size");
int choice = scan.nextInt();
switch (choice){
case 1 : System.out.println("Enter the element to be inserted");
```

```
list.insertAtStart( scan.nextInt() );
break;
case 2 : System.out.println("Enter the element to be inserted");
list.insertAtEnd( scan.nextInt() );
break;
case 3 : System.out.println("Enter the element to be inserted");
int num = scan.nextInt() ;
System.out.println("Enter the position");
int pos = scan.nextInt() ;
if (pos <= 1 || pos > list.getSize() )
System.out.println("Invalid position\n");
else list.insertAtPos(num, pos);
break;
case 4 : System.out.println("Enter the position");
int p = scan.nextInt() ;
if (p < 1 || p > list.getSize()) System.out.println("Invalid position\n");
else
list.deleteAtPos(p);
break;
case 5 :
System.out.println("Empty status = "+ list.isEmpty());
break;
case 6 :
System.out.println("Size = "+ list.getSize() +" \n");
break;
default : System.out.println("Incorrect entry \n ");
break;
}
list.display();
System.out.println("\nPress 'C' to continue and 'S' to stop \n");
ch = scan.next().charAt(0);
} while (ch == 'C');
}}
```

Output :-

```
<terminated> Question1 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe
KHAN MOHD OWAIS RAZA (20BCD7138)
Lab-3 (17-09-2022)
Java code to implement single linked list and its operations

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
1
Enter the element to be inserted
10

Single Linked List: 10
Press 'C' to continue and 'S' to stop

C

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
1
Enter the element to be inserted
20

Single Linked List: 20 --> 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
1
Enter the element to be inserted
30

Single Linked List: 30 --> 20 --> 10

Press 'C' to continue and 'S' to stop

C
```

```
Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
1
Enter the element to be inserted
40

Single Linked List: 40 --> 30 --> 20 --> 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
2
Enter the element to be inserted
50
Single Linked List: 40 --> 30 --> 20 --> 10 --> 50

Press 'C' to continue and 'S' to stop

C

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
3
Enter the element to be inserted
60
Enter the position
3
Single Linked List: 40 --> 30 --> 60 --> 20 --> 10 --> 50

Press 'C' to continue and 'S' to stop

C

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
4
Enter the position
2

Single Linked List: 40 --> 60 --> 20 --> 10 --> 50

Press 'C' to continue and 'S' to stop

C
```

```
Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
5
Empty status = false

Single Linked List: 40 --> 60 --> 20 --> 10 --> 50

Press 'C' to continue and 'S' to stop

C

Please select the operation:

[1] Insert element at beginning
[2] Insert element at ending
[3] Insert element at a position
[4] Delete element at a position
[5] Check if the list is empty
[6] Display size
6
Size = 5


Single Linked List: 40 --> 60 --> 20 --> 10 --> 50

Press 'C' to continue and 'S' to stop

S
```

Q.2] Write a program to implement stack operation using linked list

```
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
/* Lab-3 (17-09-2022)*/
/* Java code to implement stack operation using linked list */
package CSE2001_Lab3_20BCD7138;
import java.util.*;
class Node1{
protected int data;
protected Node1 link;
public Node1(){
link = null;
data = 0;
}
public Node1(int d,Node1 n){
data = d;
link = n;
}
public void setLink(Node1 n){
link = n;
}
public void setData(int d){
data = d;
}
public Node1 getLink(){
return link;
}
public int getData(){
return data;
}}
class linkedStack{
protected Node1 top ;
protected int size ;
public linkedStack(){
top = null;
size = 0;
}
public boolean isEmpty(){
return top == null;
}
public int getSize(){
return size;
}
public void push(int data){
Node1 nptr = new Node1 (data, null);
if (top == null) top = nptr;
else{
nptr.setLink(top);
```

```
top = nptr;
}
size++ ;
}
public int pop(){
if (isEmpty()) throw new NoSuchElementException("Underflow Exception") ;
Node1 ptr = top;
top = ptr.getLink();
size-- ;
return ptr.getData();
}
public int peek(){
if (isEmpty()) throw new NoSuchElementException("Underflow Exception") ;
return top.getData();
}
public void display(){
System.out.print("\nStack: ");
if (size == 0){
System.out.print("Empty\n");
return ;
}
Node1 ptr = top;
while (ptr != null){
System.out.print(ptr.getData()+" ");
ptr = ptr.getLink();
}
System.out.println();
}}
public class Question2{
public static void main(String[] args){
Scanner scan = new Scanner(System.in);
linkedStack ls = new linkedStack();
System.out.println("KHAN MOHD OWAIS RAZA (20BCD7138)");
System.out.println("Lab-3 (17-09-2022)");
System.out.println("Java code to implement stack operation using linked
list");
char ch;
do {
System.out.println("\nPlease select the operation:");
System.out.println("[1] Push element");
System.out.println("[2] Pop element");
System.out.println("[3] Peek element");
System.out.println("[4] Check if empty");
System.out.println("[5] Display the size");
int choice = scan.nextInt();
switch (choice){
case 1 : System.out.println("Enter element");
ls.push( scan.nextInt() );
break;
case 2 :
try{ System.out.println("Popped Element = "+ ls.pop());}
catch (Exception e){
System.out.println("Error : " + e.getMessage());
}
```

```
break;
case 3 :
try{
System.out.println("Peek Element = "+ ls.peek());
}
catch (Exception e){
System.out.println("Error : " + e.getMessage());
}
break;
case 4 : System.out.println("Empty status : "+ ls.isEmpty());
break;
case 5 : System.out.println("Size = "+ ls.getSize());
break;
case 6 : System.out.println("Stack = ");
ls.display();
break;
default : System.out.println("Incorrect entry \n ");
break;
}
ls.display();
System.out.println("\nPress 'C' to continue and 'S' to stop \n");
ch = scan.next().charAt(0);
} while (ch == 'C');
}}
```

Output :-

```
<terminated> Question2 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe
KHAN MOHD OWAIS RAZA (20BCD7138)
Lab-3 (17-09-2022)
Java code to implement stack operation using linked list

Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
1
Enter element
10

Stack: 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
1
Enter element
20

Stack: 20 10

Press 'C' to continue and 'S' to stop

C
```

```
Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
1
Enter element
30

Stack: 30 20 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
2
Popped Element = 30

Stack: 20 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
3
Peek Element = 20

Stack: 20 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
4
Empty status : false

Stack: 20 10

Press 'C' to continue and 'S' to stop

C

Please select the operation:
[1] Push element
[2] Pop element
[3] Peek element
[4] Check if empty
[5] Display the size
5
Size = 2
```

```
Stack: 20 10

Press 'C' to continue and 'S' to stop

S
```

Q.3] Write a program to implement queue operation using linked list

```
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
/* Lab-3 (17-09-2022)*/
/* Java code to implement queue operation using linked list */
package CSE2001_Lab3_20BCD7138;
import java.util.*;
class Node3{
protected int data;
protected Node3 link;
public Node3(){
link = null;
data = 0;
}
public Node3(int d,Node3 n){
data = d;
link = n;
}
public void setLink(Node3 n){
link = n;
}
public void setData(int d){
data = d;
}
public Node3 getLink(){
return link;
}
public int getData(){
return data;
}}
class Queue{
protected Node3 front, rear;
public int size;
public Queue(){
front = null;
rear = null;
size = 0;
}
public boolean isEmpty(){
return front == null;
}
public int getSize(){
return size;
}
public void insert(int data){
```

```
Node3 nptr = new Node3(data, null);
if (rear == null){
front = nptr;
rear = nptr;
}
else{
rear.setLink(nptr);
rear = rear.getLink();
}
size++ ;
}
public int remove(){
if (isEmpty()) throw new NoSuchElementException("Underflow Exception");
Node3 ptr = front;
front = ptr.getLink();
if (front == null) rear = null;
size-- ;
return ptr.getData();
}
public int peek(){
if (isEmpty()) throw new NoSuchElementException("Underflow Exception");
return front.getData();
}
public void display(){
System.out.print("\nQueue: ");
if (size == 0){
System.out.print("Empty\n");
return ;
}
Node3 ptr = front;
while (ptr != rear.getLink()){
System.out.print(ptr.getData()+" ");
ptr = ptr.getLink();
}
System.out.println();
}}
public class Question3{
public static void main(String[] args){
Scanner scan = new Scanner(System.in);
Queue lq = new Queue();
System.out.println("KHAN MOHD OWAIS RAZA (20BCD7138)");
System.out.println("Lab-3 (17-09-2022)");
System.out.println("Java code to implement queue operation using linked list");
char ch;
do{
System.out.println("\nPlease select operations:");
System.out.println("[1] Insert element");
System.out.println("[2] Remove element");
System.out.println("[3] Peek element");
System.out.println("[4] Check if empty");
System.out.println("[5] Display size");
int choice = scan.nextInt();
switch (choice){
case 1 : System.out.println("Enter element");
lq.insert( scan.nextInt() );
break;
case 2 :
try{
System.out.println("Removed element: "+ lq.remove());
```

```
}
catch (Exception e){
System.out.println("Error : " + e.getMessage());
}
break;
case 3 :
try{
System.out.println("Peeked element: "+ lq.peek());
}
catch (Exception e){
System.out.println("Error : " + e.getMessage());
}
break;
case 4 : System.out.println("Empty status: "+ lq.isEmpty());
break;
case 5 : System.out.println("Size: "+ lq.getSize());
break;
default : System.out.println("Incorrect entry \n ");
break;
}
lq.display();
System.out.println("\nPress 'C' to continue and 'S' to stop \n");
ch = scan.next().charAt(0);
} while (ch == 'C');
}}
```

Output :-

```
<terminated> Question3 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe
KHAN MOHD OWAIS RAZA (20BCD7138)
Lab-3 (17-09-2022)
Java code to implement queue operation using linked list

Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
1
Enter element
10

Queue: 10

Press 'C' to continue and 'S' to stop

C

Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
1
Enter element
20
```

```
Queue: 10 20

Press 'C' to continue and 'S' to stop

C

Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
1
Enter element
30

Queue: 10 20 30

Press 'C' to continue and 'S' to stop

C

Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
2
Removed element: 10

Queue: 20 30

Press 'C' to continue and 'S' to stop

C

Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
3
Peeked element: 20

Queue: 20 30

Press 'C' to continue and 'S' to stop

C

Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
4
Empty status: false

Queue: 20 30

Press 'C' to continue and 'S' to stop

C
```

```
Please select operations:
[1] Insert element
[2] Remove element
[3] Peek element
[4] Check if empty
[5] Display size
5
Size: 2

Queue: 20 30

Press 'C' to continue and 'S' to stop

S
```