Bahria University,

Karachi Campus

A picture containing text, room

Description automatically generated

LAB EXPERIMENT NO.

\_\_\_\_**3**\_\_\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | **Write a program to create a linked list and perform**  **\*traversing**  **\* Insertion**  **\*deletion.** |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

\_\_\_\_4/11/2021\_\_\_\_

(Date: DD/MM/YY)

**Task No. 1 : Write a program to create a linked list and perform**

**\* traversing \* Insertion \* deletion.**

**Solution:**

**Node Class**

public class Node

{

public int data;

public Node next;

public Node(int data){

this.data = data;

next = null;

}

}

**Linked\_List Class**

class Linked\_List

{

public Node Head;

public void printList() // **traversing**

{

for (Node n = Head; n!=null; n=n.next)

{

Console.Write(n.data+" ");

}

}

public void push(int data) // **Insertion at front**

{

Node new\_Node = new Node(data);

new\_Node.next = Head;

Head = new\_Node;

}

public void insertAfter(int data , Node insertAfter) // **Insertion after a given node**

{

if(insertAfter.next == null)

{

Console.WriteLine("The given previous node cannot be null");

return;

}

Node new\_Node = new Node(data);

new\_Node.next = insertAfter.next;

insertAfter.next = new\_Node;

}

public void append(int data) // **Insertion at the End**

{

Node new\_Node = new Node(data);

if(Head == null)

{

Head = new\_Node;

return;

}

for(Node last = Head; last.next!=null||last.next==null; last= last.next)

{

if (last.next == null) {

last.next = new\_Node;

break; }

}

}

public void deleteNode(int key) // **Deletion**

{

Node temp = Head, prev = null;

if (temp != null && temp.data == key)

{

Head = temp.next;

return;

}

while (temp != null && temp.data != key)

{

prev = temp;

temp = temp.next;

}

if (temp == null)

return;

prev.next = temp.next;

}

}

**Main Method**

static void Main(string[] args)

{

Linked\_List linkList = new Linked\_List();

linkList.Head = new Node(1);

Node second = new Node(2);

Node third = new Node(3);

linkList.Head.next = second; // Link first node with the second node

second.next = third; // Link second node with the third node

Console.WriteLine("Traversing a Link List :");

linkList.printList();

Console.WriteLine("\n\nINSERTION-At the front of the linked list :");

linkList.push(0);

linkList.printList();

Console.WriteLine("\n\nINSERTION-Add a node at the end :");

linkList.append(4);

linkList.printList();

Console.WriteLine("\n\nINSERTION-Add a node after a given node :");

linkList.insertAfter(200, second);

linkList.printList();

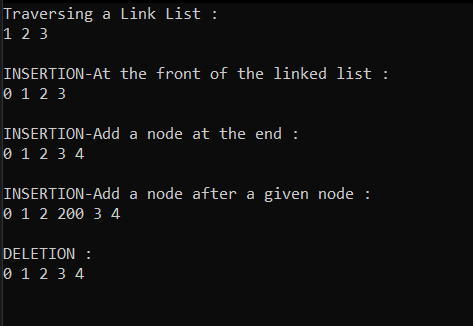
Console.WriteLine("\n\nDELETION :");

linkList.deleteNode(200);

linkList.printList();

Console.ReadLine();

}



**Output:**