

**LAB-06**

**Title: Algorithm to implement insert and delete operations on Linked List.**

**Name: Azizul Abedin Azmi**

**ID: 2022-1-60-130**

**Section: 03**

**Course Code: CSE207**

**Course Title: (Data Structures)**

**Date: 18/03/2024**

**Course Instructor:**

**Dr. Anup Kumar Paul**

**Associate Professor**

**Department of Computer Science and Engineering**

**Source Code:**

**LinkedList.java:**

package Lab06;

public class LinkedList {

    Node start;

    LinkedList() {

        start = null;

    }

    // Insertion at the beginning

    void insertBegin(int data) {

        Node newNode = new Node(data);

        if(start == null) {

            start = newNode;

            return;

        }

        else {

            newNode.next = start;

            start = newNode;

        }

    }

    // Insertion at the middle after a specified nodea

    void insertMiddle(int data, int afterData) {

        Node newNode = new Node(data);

        Node p = start;

        while (p != null && p.data != afterData) {

            p = p.next;

        }

        if (p != null) {

            newNode.next = p.next;

            p.next = newNode;

        } else {

            System.out.println("Node with specified data not found.");

        }

    }

    // Insertion at the end

    void insertEnd(int data) {

        Node newNode = new Node(data);

        if (start == null) {

            start = newNode;

            return;

        }

        Node p = start;

        while (p.next != null) {

            p = p.next;

        }

        p.next = newNode;

    }

    // Deletion of the first node

    void deleteFirst() {

        if (start == null) {

            System.out.println("List is empty. Nothing to delete.");

            return;

        }

        start = start.next;

    }

    // Deletion of a specified node

    void deleteMiddle(int data) {

        Node p = start;

        Node prev = null;

        if (p != null && p.data == data) {

            start = p.next;

            return;

        }

        while (p != null && p.data != data) {

            prev = p;

            p = p.next;

        }

        if (p == null) {

            System.out.println("Node with specified data not found.");

            return;

        }

        prev.next = p.next;

    }

    // Deletion of the last node

    void deleteLast() {

        if (start == null) {

            System.out.println("List is empty. Nothing to delete.");

            return;

        }

        Node p = start;

        Node prev = null;

        while (p.next != null) {

            prev = p;

            p = p.next;

        }

        if (prev == null) {

            start = null;

        } else {

            prev.next = null;

        }

    }

    // Traversal

    void traverse() {

        Node p = start;

        while (p != null) {

            System.out.print(p.data + " ");

            p = p.next;

        }

        System.out.println();

    }

}

**Main.java:**

package Lab06;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        LinkedList list = new LinkedList();

        int choice, data, afterData, deleteData;

        do {

            System.out.println("\n1. Insert at the beginning");

            System.out.println("2. Insert at the middle");

            System.out.println("3. Insert at the end");

            System.out.println("4. Delete first");

            System.out.println("5. Delete middle");

            System.out.println("6. Delete last");

            System.out.println("7. Traverse");

            System.out.println("8. Exit");

            System.out.print("Enter your choice: ");

            choice = scanner.nextInt();

            switch (choice) {

                case 1:

                    System.out.print("Enter data to insert at the beginning: ");

                    data = scanner.nextInt();

                    list.insertBegin(data);

                    break;

                case 2:

                    System.out.print("Enter data to insert: ");

                    data = scanner.nextInt();

                    System.out.print("Enter the data of the node after which to insert: ");

                    afterData = scanner.nextInt();

                    list.insertMiddle(data, afterData);

                    break;

                case 3:

                    System.out.print("Enter data to insert at the end: ");

                    data = scanner.nextInt();

                    list.insertEnd(data);

                    break;

                case 4:

                    list.deleteFirst();

                    break;

                case 5:

                    System.out.print("Enter data to delete: ");

                    deleteData = scanner.nextInt();

                    list.deleteMiddle(deleteData);

                    break;

                case 6:

                    list.deleteLast();

                    break;

                case 7:

                    System.out.println("Linked list elements:");

                    list.traverse();

                    break;

                case 8:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Invalid choice!");

            }

        } while (choice != 8);

        scanner.close();

    }

}

**Node.java:**

package Lab06;

public class Node {

    int data;

    Node next;

    Node(int data) {

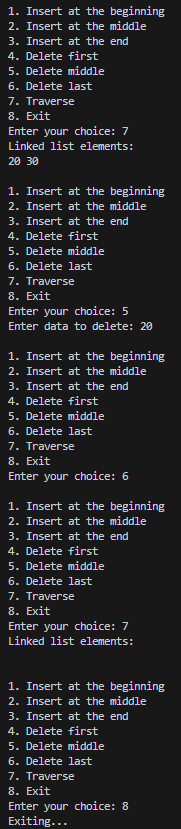
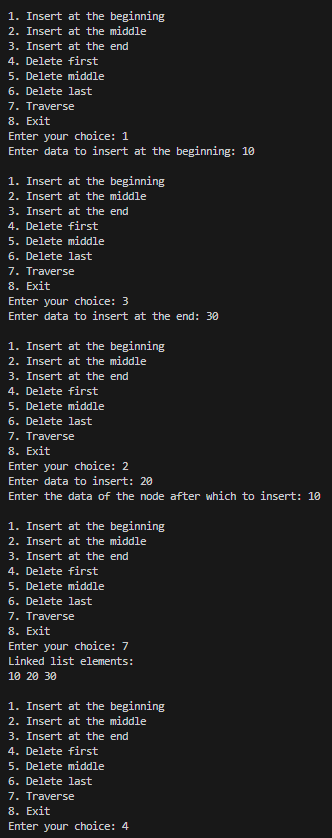
        this.data = data;

        this.next = null;

    }

}

**Output:**

****