Personal Information

Name: Khan Muhammad

CMS ID: 023-22-0199

**Assignment 2**

**Source Code:**

class Node

{

    int data;

    Node next;

    Node(int data)

    {

        this.data = data;

        this.next = null;

    }

}

class LinkedList

{

    Node head;

    Node tail;

    LinkedList()

    {

        this.head = null;

        this.tail = null;

    }

    LinkedList(int data)

    {

        Node newNode = new Node(data);

        head = newNode;

        tail = newNode;

    }

    void addAfter(Node afterNode, int data)

    {

        if(head==null || afterNode==null)

        {

            System.out.println("Error: Linked List is Full or the Given Node is Null!");

        }

        Node newNode = new Node(data);

        newNode.next = afterNode.next;

        afterNode.next = newNode;

    }

    void removeAfter(Node afterNode)

    {

        if(afterNode.next == null)

        {

            System.out.print("No Node Found After Given Node!");

            return;

        }

        afterNode.next = afterNode.next.next;

    }

    void addBefore(Node beforeNode, int data)

    {

        if(head==null || beforeNode==null)

        {

            System.out.println("Error: Linked List is Full or the Given Node is Null!");

        }

        Node newNode = new Node(data);

        Node temp = head;

        if(beforeNode.data==head.data && beforeNode.next==head.next)        // In case the given node is the first one

        {

            newNode.next = head;

            head = newNode;

            return;

        }

        while(temp!=null)

        {

            if(beforeNode.data==temp.next.data && beforeNode.next==temp.next.next)

            {

                newNode.next = temp.next;

                temp.next = newNode;

                return;

            }

            temp = temp.next;

        }

    }

    void removeBefore(Node beforeNode)

    {

        if(head == beforeNode)

        {

            System.out.print("No Node Found Before the Given Node!\n");

            return;

        }

        Node temp = head;

        Node prev = head;

        while(temp!=null)

        {

            if(beforeNode.data==temp.next.data && beforeNode.next==temp.next.next)

            {

                prev.next = prev.next.next;

                return;

            }

            prev = temp;

            temp = temp.next;

        }

    }

    void updateElementAfter(Node afterNode, int data)

    {

        if(head==null)

        {

            System.out.print("Stack is Empty!");

            return;

        }

        if(afterNode.next==null)

        {

            System.out.print("No Node Found After Given Node!");

            return;

        }

        afterNode.next.data = data;

    }

    void updateElementBefore(Node beforeNode, int data)

    {

        if(head==null)

        {

            System.out.print("Stack is Empty!");

            return;

        }

        if(head==beforeNode)

        {

            System.out.print("No Node Found Before Given Node!");

            return;

        }

        Node temp = head;

        while(temp!=null)

        {

            if(beforeNode.data==temp.next.data && beforeNode.next==temp.next.next)

            {

                temp.data = data;

                return;

            }

            temp = temp.next;

        }

    }

    void printList()                // Print List

    {

        Node temp = head;

        System.out.print("{");

        while(temp!=null)

        {

            System.out.print(temp.data+", ");

            temp = temp.next;

        }

        System.out.println("\b\b}");

    }

}

class Assignment2

{

    public static void main(String [] args)

    {

        LinkedList l1 = new LinkedList(10);

        Node second = new Node(20);

        Node third = new Node(30);

        Node forth = new Node(40);

        l1.head.next = second;

        second.next = third;

        third.next = forth;

        System.out.print("\nList: ");

        l1.printList();

        System.out.print("\n1. Adding 35 After Third Node: ");

        l1.addAfter(third, 35);

        l1.printList();

        System.out.print("\n2. Adding 15 Before Second Node: ");

        l1.addBefore(second, 15);

        l1.printList();

        System.out.print("\n3. Removing After Forth Node: ");

        l1.removeAfter(third);      // 'third' because the Forth one in the list now, is third

        l1.printList();

        System.out.print("\n4. Removing Before Third Node: ");

        l1.removeBefore(second);      // 'second' because the Third one in the list now, is second

        l1.printList();

        System.out.print("\n5. Updating Element To '35' After Second Node: ");

        l1.updateElementAfter(second, 25);

        l1.printList();

        System.out.print("\n6. Updating Element To '15' Before Second Node: ");

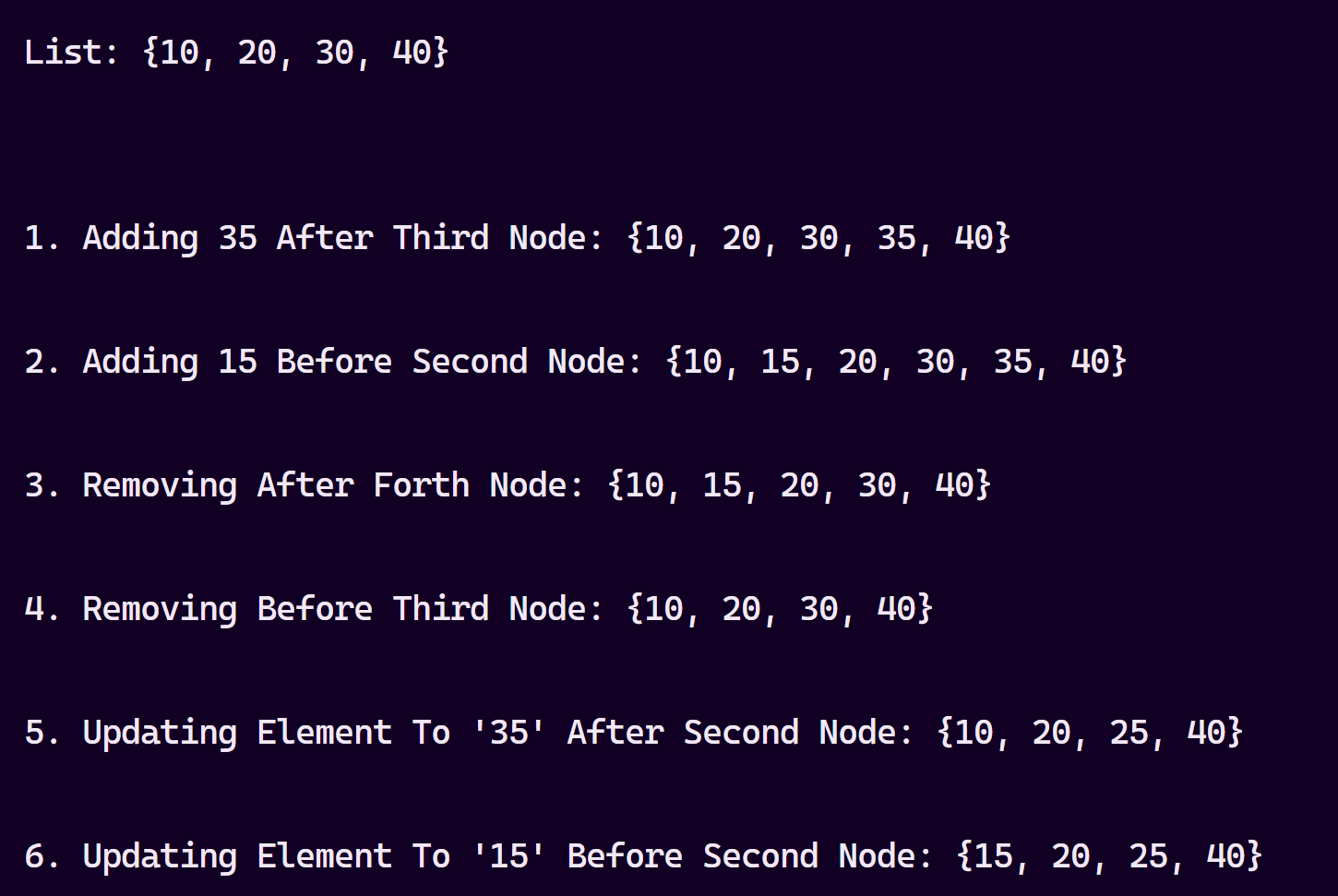
        l1.updateElementBefore(second, 15);

        l1.printList();

    }

}

**Result:**

****