**Assisted Practice: 4.5 Singly Linked List**

This section will guide you to:

* Write a program in Java to delete the first occurrence of a key in a singly linked list
* Use Eclipse (the popular text editor for Java programs)
* Push code to Git

**1.Writing the program in Java to understand the working of the singly linked list**

public class SinglyLinkedList {

    //Represent a node of the singly linked list

    class Node{

        int data;

        Node next;

        public Node(int data) {

            this.data = data;

            this.next = null;

        }

    }

    //Represent the head and tail of the singly linked list

    public Node head = null;

    public Node tail = null;

    //addNode() will add a new node to the list

    public void addNode(int data) {

        //Create a new node

        Node newNode = new Node(data);

        //Checks if the list is empty

        if(head == null) {

            //If list is empty, both head and tail will point to new node

            head = newNode;

            tail = newNode;

        }

        else {

            //newNode will be added after tail such that tail's next will point to newNode

            tail.next = newNode;

            //newNode will become new tail of the list

            tail = newNode;

        }

    }

    //display() will display all the nodes present in the list

    public void display() {

        //Node current will point to head

        Node current = head;

        if(head == null) {

            System.out.println("List is empty");

            return;

        }

        System.out.println("Nodes of singly linked list: ");

        while(current != null) {

            //Prints each node by incrementing pointer

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

            current = current.next;

        }

        System.out.println();

    }

    public static void main(String[] args) {

        SinglyLinkedList sList = new SinglyLinkedList();

        //Add nodes to the list

        sList.addNode(1);

        sList.addNode(2);

        sList.addNode(3);

        sList.addNode(4);

        //Displays the nodes present in the list

        sList.display();

    }

}

**Output:**

Nodes of singly linked list:

1 2 3 4