1.Singly Linked list examples in java.

Singular Linked List. The type of linked list consisting of a sequence of nodes where each node consists of data and a link to the next node, that can be traversed from the first node of the list (also called as head) to the last node of the list (also called as Tail) and is unidirectional is called Singly Linked list.

Example: A Train

2.Java program to create and display a singly linked list.

```
public class LinkedList {
	public static class Node {
		int data;
		Node next;

		public Node(int data) {
			this.data=data;
			this.next=null;
		}
	}

	public static Node head ;

	public void insert(int data) {

		Node temp =new Node(data);

			if(head == null) {
				head=temp;
				return ;
			}
				temp.next=head;
				head=temp;
			}

	public void printList() {

		if(head==null) {
			System.out.println("List is Empty");
			return ;
		}

		Node temp=head;

			while(temp!=null) {

				System.out.print(temp.data);
				if(temp.next!=null) {
```

```
                    System.out.print("->");
                }
                    temp=temp.next;
            }
                    System.out.println("");
        }
}

public class Main {
        public static void main(String[] args) {
            LinkedList list = new LinkedList();
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter No List ");
            int n= sc.nextInt();

            for(int i=1;i<=n;i++) {
                list.insert(i);
                list.printList();
            }
        }
}
```

3.Java program to create singly linked list of n nodes and count the number of nodes.

```
public class LinkedList {
        public static class Node {
                int data;
                Node next;

            public Node(int data) {
                    this.data=data;
                    this.next=null;
                }
            }

        public static Node head ;

        public void insert(int data) {

            Node temp =new Node(data);

                if(head == null) {
                    head=temp;
                    return ;
                }
                    temp.next=head;
                    head=temp;
                }
        public int countList() {
```

```
            int count=0;
            Node curr= head;

            while(curr!=null) {
                  count++;
                  curr=curr.next;
            }
            return count;
      }
}

public class Main {
      public static void main(String[] args) {
            LinkedList list = new LinkedList();
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter No List ");
            int n= sc.nextInt();

            for(int i=1;i<=n;i++) {
                  list.insert(i);
            }
System.out.println("The List Count is = "+list.countList());
      }
}
```

4.Java program to create a singly linked list of n nodes and display it in reverse order.

```
public class LinkedList {
      public static class Node {
            int data;
            Node next;

            public Node(int data) {
                  this.data=data;
                  this.next=null;
            }
      }

      public static Node head ;

      public void insert(int data) {

            Node temp =new Node(data);

                  if(head == null) {
                        head=temp;
                        return ;
            }
                  temp.next=head;
                  head=temp;
```

```
	}

	public void printList() {

		if(head==null) {
			System.out.println("List is Empty");
			return ;
		}
		Node temp=head;

		while(temp!=null) {

			System.out.print(temp.data);
			if(temp.next!=null) {
				System.out.print("->");
			}
				temp=temp.next;
		}
			System.out.println("");
	}

	public void reverse () {
		Node curr=head;
		Node prev=null;
		Node nex;
		while(curr!=null) {
			nex=curr.next;
			curr.next=prev;
			prev=curr;
			curr=nex;
		}
			head=prev;
	}
}

public class Main {
	public static void main(String[] args) {
		LinkedList list = new LinkedList();
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter No List ");
		int n= sc.nextInt();

		for(int i=1;i<=n;i++) {
			list.insert(i);

	}
	System.out.print("List before Reverse = ");
	list.printList();

	System.out.print("List After Reverse = ");
	list.reverse();
```

```
        list.printList();
        }
}
```

---

5. Java program to delete a node from the beginning of the singly linked list.

```
public class LinkedList {
        public static class Node {
                int data;
                Node next;

        public Node(int data) {
                this.data=data;
                this.next=null;

        }
}
        public static Node head ;

        public void insert(int data) {

                Node temp =new Node(data);
                        if(head == null) {
                                head=temp;
                                return ;
                }
                        temp.next=head;
                        head=temp;
}
        public int countList() {
                        int count=0;
                        Node curr= head;
                        while(curr!=null){
                                count++;
                                curr=curr.next;
                }
                        return count;
        }
}

public class Main {
        public static void main(String[] args) {
                LinkedList list = new LinkedList();
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter No List ");
                int n= sc.nextInt();

                for(int i=1;i<=n;i++) {
                        list.insert(i);
                }
```

```
        System.out.println("The First Node from List value is =
        "+ list.removeFirst());
        }
}
```