[**Linked-List | Java | Simple**](https://community.scaler.com/t/linked-list-java-simple/16365)

[Academy](https://community.scaler.com/c/academy/23)[Linked Lists](https://community.scaler.com/c/academy/linked-lists/34)

[linked-list](https://community.scaler.com/tag/linked-list)

[Feb 2022](https://community.scaler.com/t/linked-list-java-simple/16365)

**1 / 1**

Feb 2022

[Feb 2022](https://community.scaler.com/t/linked-list-java-simple/16365)

[A white letter on a green background

Description automatically generated](https://community.scaler.com/u/skmirazhaque_ec8665a)

[**skmirazhaque\_ec8665a**](https://community.scaler.com/u/skmirazhaque_ec8665a)

[Feb '22](https://community.scaler.com/t/linked-list-java-simple/16365)

static Node head;

    static int length;

    public static class Node{

        Node next;

        int val;

        Node(int val){

            this.val=val;

        }

    }

    public static void insert\_node(int position, int value) {

        // @params position, integer

        // @params value, integer

      if(position>length+1) {

          return;

      }

        Node newNode = new Node(value);

        if(position == 1){

            newNode.next = head;

            head = newNode;

        }

        else {

            int counter=1;

            Node curr = head;

            while(counter<position-1){

                curr=curr.next;

                counter++;

            }

            newNode.next=curr.next;

            curr.next=newNode;

        }

        length++;

    }

    public static void delete\_node(int position) {

        // @params position, integer

        if(position>length) {

            return ;

        }

        if(position==1){

            head=head.next;

        }else{

            int counter=1;

            Node curr=head;

            while(counter<position-1){

                counter++;

                curr=curr.next;

            }

            curr.next=curr.next.next;

        }

        length--;

    }

    public static void print\_ll() {

        // Output each element followed by a space

        if(length == 0) {

            return;

        }

        Node node=head;

        while (node.next != null) {

            System.out.print(node.val+" ");

            node = node.next;

        }

        System.out.print(node.val);

    }