**LTIMindTree**

**Day -1: 08/09/2025**

Technical:

1. Student database using Class and Object
2. String Comparession

//AAABBCCD -> A3B2C2D

    public static String compress(String str){

        int len = str.length();

        int count = 1;

        String res = "";

        for(int i = 0;i<len;i++){

            if(i+1 < len && str.charAt(i) == str.charAt(i+1)){

                count++;

            }else{

                res = res + str.charAt(i) + count;

                count = 1;

            }

        }

        return res;

    }

1. Length of the Longest SubString Without Duplicate Character

public static String subStringWDC(String s) {

        int n = s.length();

        int left = 0;

        int maxLen = 0;

        int start = 0; // track start of best substring

        Map<Character, Integer> map = new HashMap<>();

        for (int right = 0; right < n; right++) {

            char ch = s.charAt(right);

            // if char is repeated, move left pointer

            if (map.containsKey(ch) && map.get(ch) >= left) {

                left = map.get(ch) + 1;

            }

            map.put(ch, right);

            if (right - left + 1 > maxLen) {

                maxLen = right - left + 1;

                start = left;

            }

        }

        return s.substring(start, start + maxLen);

    }

1. Given a string of characters, find the largest and the smallest word

public static void findLargestAndSmallest(String str) {

        // split words by spaces

        String[] words = str.split("\\s+");

        String smallest = words[0];

        String largest = words[0];

        for (String word : words) {

            if (word.length() < smallest.length()) {

                smallest = word;

            }

            if (word.length() > largest.length()) {

                largest = word;

            }

        }

        System.out.println("Smallest word: " + smallest);

        System.out.println("Largest word: " + largest);

    }

1. Remove all Occurrences of all X character

public static void removeX(){

        String str1 = "Mokhanes Sanjay Kumar Nathis";

        String str2 = "a";

        System.out.println(str1.replaceAll(str2, ""));

    }

1. Encryption with (-3) or (+3) Characters

public static void encrp3(){

        //(Right Shift (+3) -> (char) (((c - 'A' + s) % 26) + 'A');)

        //(Left Shift (-3) -> (char) (((c - 'A' - s + 26) % 26) + 'A');)

        String text = "Sanjay";

        int s = 3;

        StringBuffer res = new StringBuffer();

        for(int i = 0;i<text.length();i++){

            char c = text.charAt(i);

            if(Character.isUpperCase(c)){

                char ch = (char) (((c - 'A' + s) % 26) + 'A');

                res.append(ch);

            }else if(Character.isLowerCase(c)){

                char ch = (char) (((c - 'a' + s) % 26) + 'a');

                res.append(ch);

            }else{

                res.append(c);

            }

        }

        System.out.println(res);

    }

1. Find Penalty: