**Strings**

**RA2411003011058**

**SHARVESWAR MADASAMY**

Q1)

public class Q1 {

    public static void main(String[] args) {

        // 1. String literal

        String strLiteral = "Java Programming";

        // 2. new String() constructor

        String strObject = new String("Java Programming");

        // 3. Character array

        char[] charArray = {'J', 'a', 'v', 'a', ' ', 'P', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g'};

        String strFromArray = new String(charArray);

        // Compare using == (reference comparison)

        System.out.println("strLiteral == strObject: " + (strLiteral == strObject));

        System.out.println("strLiteral == strFromArray: " + (strLiteral == strFromArray));

        // Compare using equals() (content comparison)

        System.out.println("strLiteral.equals(strObject): " + strLiteral.equals(strObject));

        System.out.println("strLiteral.equals(strFromArray): " + strLiteral.equals(strFromArray));

        // Escape sequences

        String quote = "Programming Quote:\n\t\"Code is poetry\" - Unknown\n\tPath: C:\\Java\\Projects";

        System.out.println("\n" + quote);

    }

}

Q2)

import java.util.Scanner;

public class Q2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        // Full name

        System.out.print("Enter your full name: ");

        String fullName = sc.nextLine();

        String[] parts = fullName.split(" ");

        String firstName = parts[0];

        String lastName = parts[parts.length - 1];

        // Favorite language

        System.out.print("Enter your favorite programming language: ");

        String favLang = sc.nextLine();

        // Sentence

        System.out.print("Enter a sentence about your programming experience: ");

        String sentence = sc.nextLine();

        // Character count (excluding spaces)

        int charCount = sentence.replace(" ", "").length();

        // Output

        System.out.println("\n=== SUMMARY ===");

        System.out.println("First Name: " + firstName);

        System.out.println("Last Name: " + lastName);

        System.out.println("Favorite Language: " + favLang.toUpperCase());

        System.out.println("Character count (no spaces): " + charCount);

        sc.close();

    }

}

Q3)

public class Q3 {

    public static String findLongestName(String[] names) {

        String longest = names[0];

        for (String name : names) {

            if (name.length() > longest.length()) {

                longest = name;

            }

        }

        return longest;

    }

    public static int countNamesStartingWith(String[] names, char letter) {

        int count = 0;

        letter = Character.toLowerCase(letter);

        for (String name : names) {

            if (Character.toLowerCase(name.charAt(0)) == letter) {

                count++;

            }

        }

        return count;

    }

    public static void main(String[] args) {

        String[] students = {"John Smith", "Alice Johnson", "Bob Brown", "Carol Davis", "David Wilson"};

        System.out.println("Longest Name: " + findLongestName(students));

        System.out.println("Names starting with 'A': " + countNamesStartingWith(students, 'A'));

        System.out.println("\nFormatted Names:");

        for (String name : students) {

            String[] parts = name.split(" ");

            System.out.println(parts[1] + ", " + parts[0]);

        }

    }

}

Q4)

import java.util.Arrays;

import java.util.Scanner;

public class Q4 {

    public static String cleanInput(String text) {

        // Remove spaces from start/end and reduce multiple spaces to one

        String cleaned = text.trim();

        String result = "";

        boolean spaceFound = false;

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

            char ch = cleaned.charAt(i);

            if (ch != ' ') {

                result += ch;

                spaceFound = false;

            } else {

                if (!spaceFound) { // avoid multiple spaces

                    result += ' ';

                    spaceFound = true;

                }

            }

        }

        return result;

    }

    public static void analyzeText(String text) {

        String[] words = text.split(" ");

        int wordCount = words.length;

        // Count characters without spaces

        int charCount = 0;

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

            if (text.charAt(i) != ' ') {

                charCount++;

            }

        }

        // Longest word

        String longest = words[0];

        for (String w : words) {

            if (w.length() > longest.length()) {

                longest = w;

            }

        }

        // Most common character

        char mostCommon = ' ';

        int maxCount = 0;

        for (char c = 'a'; c <= 'z'; c++) {

            int count = 0;

            for (char ch : text.toLowerCase().toCharArray()) {

                if (ch == c) count++;

            }

            if (count > maxCount) {

                maxCount = count;

                mostCommon = c;

            }

        }

        System.out.println("\n=== ANALYSIS ===");

        System.out.println("Words: " + wordCount);

        System.out.println("Characters (no spaces): " + charCount);

        System.out.println("Longest Word: " + longest);

        System.out.println("Most Common Character: " + mostCommon);

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("=== TEXT PROCESSOR ===");

        System.out.print("Enter a paragraph: ");

        String text = sc.nextLine();

        text = cleanInput(text);

        analyzeText(text);

        // Sorted words

        String[] words = text.toLowerCase().split(" ");

        Arrays.sort(words);

        System.out.println("\nWords in alphabetical order: " + Arrays.toString(words));

        // Search word

        System.out.print("\nEnter a word to search: ");

        String search = sc.nextLine().toLowerCase();

        boolean found = false;

        for (String w : words) {

            if (w.equals(search)) {

                found = true;

                break;

            }

        }

        System.out.println(found ? "Word found!" : "Word not found.");

        sc.close();

    }

}