LEVEL 3 PRACTICE PROGRAM

1. **An organization took up the exercise to find the Body Mass Index (BMI) of all the persons in a team of 10 members. For this create a program to find the BMI and display the height, weight, BMI, and status of each individual**

```java
import java.util.Scanner;
public class BMICalculator {

    // Method to calculate BMI and determine status
    public static String[] calculateBMIStatus(double weight, double height) {
        // Convert height from cm to meters
        double heightInMeters = height / 100;
        double bmi = weight / (heightInMeters * heightInMeters);

        String status;
        if (bmi <= 18.4) {
            status = "Underweight";
        } else if (bmi >= 18.5 && bmi <= 24.9) {
            status = "Normal";
        } else if (bmi >= 25.0 && bmi <= 39.9) {
            status = "Overweight";
        } else {
            status = "Obese";
        }

        return new String[] {
            String.format("%.1f cm", height),
            String.format("%.1f kg", weight),
            String.format("%.2f", bmi),
            status
        };
    }

    // Method to process all team members' data
    public static String[][] processTeamData(double[][] heightWeightData) {
        String[][] results = new String[10][4];

        for (int i = 0; i < heightWeightData.length; i++) {
            double weight = heightWeightData[i][0];
            double height = heightWeightData[i][1];
            results[i] = calculateBMIStatus(weight, height);
        }

        return results;
    }
```

```java
    // Method to display results in tabular format
    public static void displayResults(String[][] bmiData) {
        System.out.println("\nTeam BMI Report:");
        System.out.println("+---------+------------+------------+------------+--------------+");
        System.out.println("| Person  | Height     | Weight     | BMI        | Status       |");
        System.out.println("+---------+------------+------------+------------+--------------+");

        for (int i = 0; i < bmiData.length; i++) {
            System.out.printf("| %7d | %10s | %10s | %10s | %12s |\n",
                              (i+1),
                              bmiData[i][0],
                              bmiData[i][1],
                              bmiData[i][2],
                              bmiData[i][3]);
        }

        System.out.println("+---------+------------+------------+------------+--------------+");
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        final int TEAM_SIZE = 10;
        double[][] heightWeightData = new double[TEAM_SIZE][2];

        System.out.println("Enter height (cm) and weight (kg) for 10 team members:");

        for (int i = 0; i < TEAM_SIZE; i++) {
            System.out.println("\nMember " + (i+1) + ":");
            System.out.print("Height in cm: ");
            heightWeightData[i][1] = scanner.nextDouble();
            System.out.print("Weight in kg: ");
            heightWeightData[i][0] = scanner.nextDouble();
        }

        String[][] bmiResults = processTeamData(heightWeightData);
        displayResults(bmiResults);

        scanner.close();
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac BMICalculator.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java BMICalculator
Enter height (cm) and weight (kg) for 10 team members:

Member 1:
Height in cm: 169
Weight in kg: 65

Member 2:
Height in cm: 171
Weight in kg: 68

Member 3:
Height in cm: 154
Weight in kg: 55

Member 4:
Height in cm: 141
Weight in kg: 95

Member 5:
Height in cm: 166
Weight in kg: 59

Member 6:
Height in cm: 167
Weight in kg: 70

Member 7:
Height in cm: 175
Weight in kg: 84

Member 8:
Height in cm: 168
Weight in kg: 68

Member 9:
Height in cm: 161
Weight in kg: 59

Member 10:
Height in cm: 166
Weight in kg: 78

Team BMI Report:
+---------+------------+------------+------------+--------------+
| Person  | Height     | Weight     | BMI        | Status       |
+---------+------------+------------+------------+--------------+
|       1 |   169.0 cm |    65.0 kg |      22.76 |       Normal |
|       2 |   171.0 cm |    68.0 kg |      23.26 |       Normal |
|       3 |   154.0 cm |    55.0 kg |      23.19 |       Normal |
|       4 |   141.0 cm |    95.0 kg |      47.78 |        Obese |
|       5 |   166.0 cm |    59.0 kg |      21.41 |       Normal |
|       6 |   167.0 cm |    70.0 kg |      25.10 |   Overweight |
|       7 |   175.0 cm |    84.0 kg |      27.43 |   Overweight |
|       8 |   168.0 cm |    68.0 kg |      24.09 |       Normal |
|       9 |   161.0 cm |    59.0 kg |      22.76 |       Normal |
|      10 |   166.0 cm |    78.0 kg |      28.31 |   Overweight |
+---------+------------+------------+------------+--------------+
```

2. **Find unique characters in a string using the charAt() method and display the result**

```java
import java.util.Scanner;
public class UniqueCharacters {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        System.out.print("Unique characters: ");
        for (char c : findUniqueChars(input)) {
            if (c != '\0') System.out.print(c + " ");
        }
    }

    public static char[] findUniqueChars(String str) {
        char[] unique = new char[str.length()];
        int uniqueCount = 0;

        for (int i = 0; i < str.length(); i++) {
            char current = str.charAt(i);
            boolean isUnique = true;

            for (int j = 0; j < uniqueCount; j++) {
                if (unique[j] == current) {
                    isUnique = false;
                    break;
                }
            }

            if (isUnique) {
                unique[uniqueCount++] = current;
            }
        }

        return unique;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac UniqueCharacters.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java UniqueCharacters
Enter a string: Shounak
Unique characters: S h o u n a k
```

3. **Write a program to find the first non-repeating character in a string and show the result**

```java
import java.util.Scanner;
public class FirstNonRepeating {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        char result = findFirstNonRepeating(input);
        System.out.println(result != '\0' ?
            "First non-repeating character: " + result :
            "No non-repeating characters found");
    }

    public static char findFirstNonRepeating(String str) {
        int[] freq = new int[256];

        for (int i = 0; i < str.length(); i++) {
            freq[str.charAt(i)]++;
        }

        for (int i = 0; i < str.length(); i++) {
            if (freq[str.charAt(i)] == 1) {
                return str.charAt(i);
            }
        }

        return '\0';
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac FirstNonRepeating.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java FirstNonRepeating
Enter a string: Shounak
First non-repeating character: S
```

4. **Write a program to find the frequency of characters in a string using the charAt() method and display the result**

```java
import java.util.Scanner;
public class CharacterFrequency {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String[][] frequencyTable = getCharacterFrequency(input);

        System.out.println("\nCharacter Frequency Table:");
        System.out.println("+------------+------------+");
        System.out.println("| Character  | Frequency  |");
        System.out.println("+------------+------------+");

        for (String[] row : frequencyTable) {
            if (row != null) {
                System.out.printf("| %-10s | %-10s |\n", row[0], row[1]);
            }
        }

        System.out.println("+------------+------------+");

        scanner.close();
    }

    public static String[][] getCharacterFrequency(String str) {
        // Array to store frequencies (using ASCII values as indices)
        int[] frequency = new int[256];

        // Count character frequencies
        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            frequency[currentChar]++;
        }

        // Count how many unique characters we have
        int uniqueCount = 0;
        for (int count : frequency) {
            if (count > 0) {
                uniqueCount++;
            }
        }
```

```java
        // Create 2D array to store results
        String[][] result = new String[uniqueCount][2];
        int index = 0;

        // Fill the result array
        for (int i = 0; i < frequency.length; i++) {
            if (frequency[i] > 0) {
                result[index][0] = getPrintableChar(i); // Character
                result[index][1] = String.valueOf(frequency[i]); // Frequency
                index++;
            }
        }

        return result;
    }

    // Helper method to handle special characters
    private static String getPrintableChar(int asciiValue) {
        if (asciiValue == ' ') {
            return "' ' (space)";
        } else if (asciiValue == '\t') {
            return "'\\t' (tab)";
        } else if (asciiValue == '\n') {
            return "'\\n' (newline)";
        } else if (asciiValue == '\r') {
            return "'\\r' (return)";
        } else {
            return "'" + (char)asciiValue + "'";
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac CharacterFrequency.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java CharacterFrequency
Enter a string: Shounak

Character Frequency Table:
+------------+------------+
| Character  | Frequency  |
+------------+------------+
| ' ' (space) | 1          |
| 'S'        | 1          |
| 'a'        | 1          |
| 'h'        | 1          |
| 'k'        | 1          |
| 'n'        | 1          |
| 'o'        | 1          |
| 'u'        | 1          |
+------------+------------+
```

5. **Write a program to find the frequency of characters in a string using unique characters and display the result**

```java
import java.util.Scanner;
public class CharacterFrequencyWithUnique {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        char[] uniqueChars = findUniqueCharacters(input);
        String[][] frequencyTable = getCharacterFrequency(input, uniqueChars);

        System.out.println("\nCharacter Frequency Table:");
        System.out.println("+------------+------------+");
        System.out.println("| Character  | Frequency  |");
        System.out.println("+------------+------------+");

        for (String[] row : frequencyTable) {
            System.out.printf("| %-10s | %-10s |\n", row[0], row[1]);
        }

        System.out.println("+------------+------------+");

        scanner.close();
    }

    public static char[] findUniqueCharacters(String str) {
        char[] unique = new char[str.length()];
        int uniqueCount = 0;

        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            boolean isUnique = true;

            // Check if we've seen this character before
            for (int j = 0; j < uniqueCount; j++) {
                if (unique[j] == currentChar) {
                    isUnique = false;
                    break;
                }
            }

            if (isUnique) {
                unique[uniqueCount++] = currentChar;
            }
        }
```

```java
        // Trim the array to actual size
        char[] result = new char[uniqueCount];
        System.arraycopy(unique, 0, result, 0, uniqueCount);

        return result;
    }

    public static String[][] getCharacterFrequency(String str, char[] uniqueChars) {
        int[] frequency = new int[256]; // ASCII frequency counter

        // Count all character frequencies
        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            frequency[currentChar]++;
        }

        // Create result array for only unique characters
        String[][] result = new String[uniqueChars.length][2];

        for (int i = 0; i < uniqueChars.length; i++) {
            char c = uniqueChars[i];
            result[i][0] = formatCharacter(c);
            result[i][1] = String.valueOf(frequency[c]);
        }

        return result;
    }

    private static String formatCharacter(char c) {
        switch (c) {
            case ' ': return "' ' (space)";
            case '\t': return "'\\t' (tab)";
            case '\n': return "'\\n' (newline)";
            case '\r': return "'\\r' (return)";
            default: return "'" + c + "'";
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac CharacterFrequencyWithUnique.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java CharacterFrequencyWithUnique
Enter a string: racecar

Character Frequency Table:
+------------+------------+
| Character  | Frequency  |
+------------+------------+
| 'r'        | 2          |
| 'a'        | 2          |
| 'c'        | 2          |
| 'e'        | 1          |
+------------+------------+
```

6. **Write a program to find the frequency of characters in a string using nested loops and display the result**

```java
import java.util.Scanner;
public class CharacterFrequencyNestedLoops {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String[] frequencyResult = getCharacterFrequency(input);

        System.out.println("\nCharacter Frequency Table:");
        System.out.println("+------------+------------+");
        System.out.println("| Character  | Frequency  |");
        System.out.println("+------------+------------+");

        for (String entry : frequencyResult) {
            if (entry != null) {
                String[] parts = entry.split(":");
                System.out.printf("| %-10s | %-10s |\n", parts[0], parts[1]);
            }
        }

        System.out.println("+------------+------------+");

        scanner.close();
    }

    public static String[] getCharacterFrequency(String str) {
        char[] characters = str.toCharArray();
        int[] frequency = new int[characters.length];
        String[] result = new String[characters.length];
        int resultIndex = 0;

        for (int i = 0; i < characters.length; i++) {
            // Skip if we've already processed this character
            if (characters[i] == '\0') continue;

            frequency[i] = 1; // Count the current character

            // Compare with remaining characters
            for (int j = i + 1; j < characters.length; j++) {
                if (characters[i] == characters[j]) {
                    frequency[i]++;
                    characters[j] = '\0'; // Mark as processed
                }
            }
```

```java
            // Store the result for this character
            result[resultIndex++] = formatCharacter(characters[i]) + ":" + frequency[i];
        }

        // Trim the result array to remove null entries
        String[] finalResult = new String[resultIndex];
        System.arraycopy(result, 0, finalResult, 0, resultIndex);

        return finalResult;
    }

    private static String formatCharacter(char c) {
        switch (c) {
            case ' ': return "' ' (space)";
            case '\t': return "'\\t' (tab)";
            case '\n': return "'\\n' (newline)";
            case '\r': return "'\\r' (return)";
            default: return "'" + c + "'";
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac CharacterFrequencyNestedLoops.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java CharacterFrequencyNestedLoops
Enter a string: racecar

Character Frequency Table:
+------------+------------+
| Character  | Frequency  |
+------------+------------+
| 'r'        | 2          |
| 'a'        | 2          |
| 'c'        | 2          |
| 'e'        | 1          |
+------------+------------+
```

7. **Write a program to to check if a text is palindrome and display the result**

```java
import java.util.Scanner;
public class PalindromeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        System.out.println(isPalindrome(input) ?
            "The string is a palindrome" :
            "The string is not a palindrome");
    }

    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }

        return true;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac PalindromeChecker.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java PalindromeChecker
Enter a string: Shounak
The string is not a palindrome

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac PalindromeChecker.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java PalindromeChecker
Enter a string: racecar
The string is a palindrome
```

8. **Write a program to check if two texts are anagrams and display the result**

```java
import java.util.Scanner;
public class AnagramChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = scanner.nextLine();
        System.out.print("Enter second string: ");
        String str2 = scanner.nextLine();

        System.out.println(areAnagrams(str1, str2) ?
            "The strings are anagrams" :
            "The strings are not anagrams");
    }

    public static boolean areAnagrams(String str1, String str2) {
        if (str1.length() != str2.length()) return false;

        int[] freq = new int[256];

        for (int i = 0; i < str1.length(); i++) {
            freq[str1.charAt(i)]++;
            freq[str2.charAt(i)]--;
        }

        for (int count : freq) {
            if (count != 0) return false;
        }

        return true;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac AnagramChecker.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java AnagramChecker
Enter first string: Shounak
Enter second string: Roy
The strings are not anagrams
```

9. **Create a program to display a calendar for a given month and year. The program should take the month and year as input from the user and display the calendar for that month. E.g. for 07 2005 user input, the program should display the calendar**

```java
import java.util.Scanner;
public class CalendarGenerator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter month (1-12): ");
        int month = scanner.nextInt();
        System.out.print("Enter year: ");
        int year = scanner.nextInt();

        printCalendar(month, year);
    }

    public static void printCalendar(int month, int year) {
        String[] months = {"January", "February", "March", "April", "May", "June",
                           "July", "August", "September", "October", "November", "December"};
        int[] days = {31, isLeapYear(year) ? 29 : 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

        System.out.println("    " + months[month-1] + " " + year);
        System.out.println("Su Mo Tu We Th Fr Sa");

        int firstDay = getFirstDay(month, year);
        for (int i = 0; i < firstDay; i++) {
            System.out.print("   ");
        }

        for (int day = 1; day <= days[month-1]; day++) {
            System.out.printf("%2d ", day);
            if ((day + firstDay) % 7 == 0 || day == days[month-1]) {
                System.out.println();
            }
        }
    }

    public static boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }

    public static int getFirstDay(int month, int year) {
        if (month < 3) {
            month += 12;
            year--;
        }
        int k = year % 100;
        int j = year / 100;
        int day = 1;
```

```java
        // Zeller's Congruence algorithm
        int h = (day + 13*(month + 1)/5 + k + k/4 + j/4 + 5*j) % 7;
        return (h + 5) % 7; // Convert to 0=Sunday, 1=Monday, etc.
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac CalendarGenerator.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java CalendarGenerator
Enter month (1-12): 9
Enter year: 2006
   September 2006
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

10. **Write a program to create a deck of cards, initialize the deck, shuffle the deck, and distribute the deck of n cards to x number of players. Finally, print the cards the players have.**

```java
import java.util.Scanner;
import java.util.Random;
public class CardGame {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of players: ");
        int players = scanner.nextInt();
        System.out.print("Enter cards per player: ");
        int cardsPerPlayer = scanner.nextInt();

        String[] deck = initializeDeck();
        shuffleDeck(deck);
        dealCards(deck, players, cardsPerPlayer);
    }

    public static String[] initializeDeck() {
        String[] suits = {"Hearts", "Diamonds", "Clubs", "Spades"};
        String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10",
                          "Jack", "Queen", "King", "Ace"};
        String[] deck = new String[suits.length * ranks.length];

        int index = 0;
        for (String suit : suits) {
            for (String rank : ranks) {
                deck[index++] = rank + " of " + suit;
            }
        }

        return deck;
    }

    public static void shuffleDeck(String[] deck) {
        Random rand = new Random();
        for (int i = 0; i < deck.length; i++) {
            int j = rand.nextInt(deck.length);
            String temp = deck[i];
            deck[i] = deck[j];
            deck[j] = temp;
        }
    }
```

```java
    public static void dealCards(String[] deck, int players, int cardsPerPlayer) {
        if (players * cardsPerPlayer > deck.length) {
            System.out.println("Not enough cards in the deck!");
            return;
        }

        for (int p = 0; p < players; p++) {
            System.out.println("\nPlayer " + (p+1) + " cards:");
            for (int c = 0; c < cardsPerPlayer; c++) {
                System.out.println(deck[p * cardsPerPlayer + c]);
            }
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>javac CardGame.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 3>java CardGame
Enter number of players: 3
Enter cards per player: 4

Player 1 cards:
2 of Clubs
9 of Clubs
4 of Hearts
7 of Diamonds

Player 2 cards:
3 of Clubs
8 of Hearts
9 of Hearts
10 of Hearts

Player 3 cards:
6 of Spades
3 of Hearts
6 of Hearts
2 of Diamonds
```