--- Question1_StringLengthWithoutLengthMethod.java ---

```
import java.util.Scanner;

public class Question1_StringLengthWithoutLengthMethod {

    public static int getStringLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
            // Exception caught, return count
        }
        return count;
    }

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

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

        int lengthUserDefined = getStringLength(input);
        int lengthBuiltIn = input.length();

        System.out.println("Length using user-defined method: " + lengthUserDefined);
        System.out.println("Length using built-in length(): " + lengthBuiltIn);

        scanner.close();
    }
}
```

--- Question2_SplitTextWithoutSplitMethod.java ---

```
import java.util.Arrays;
import java.util.Scanner;

public class Question2_SplitTextWithoutSplitMethod {

    public static int getStringLength(String str) {
        int count = 0;
        try {
            while (true) {
```

```
            str.charAt(count);
            count++;
        }
    } catch (IndexOutOfBoundsException e) {
        // Exception caught, return count
    }
    return count;
}

public static String[] splitUsingCharAt(String text) {
    int length = getStringLength(text);
    int spaceCount = 0;

    // Count spaces
    for (int i = 0; i < length; i++) {
        if (text.charAt(i) == ' ') {
            spaceCount++;
        }
    }

    String[] words = new String[spaceCount + 1];
    int wordIndex = 0;
    int start = 0;

    for (int i = 0; i < length; i++) {
        if (text.charAt(i) == ' ') {
            words[wordIndex++] = text.substring(start, i);
            start = i + 1;
        }
    }
    // Add last word
    words[wordIndex] = text.substring(start, length);

    return words;
}

public static boolean compareStringArrays(String[] arr1, String[] arr2) {
    if (arr1.length != arr2.length) {
        return false;
    }
    for (int i = 0; i < arr1.length; i++) {
        if (!arr1[i].equals(arr2[i])) {
            return false;
        }
```

```
        }
        return true;
    }

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

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

        String[] userSplit = splitUsingCharAt(input);
        String[] builtInSplit = input.split(" ");

        System.out.println("User-defined split: " + Arrays.toString(userSplit));
        System.out.println("Built-in split: " + Arrays.toString(builtInSplit));

        boolean isEqual = compareStringArrays(userSplit, builtInSplit);
        System.out.println("Are both splits equal? " + isEqual);

        scanner.close();
    }
}

--- Question3_SplitWordsWithLengths.java ---
import java.util.Scanner;

public class Question3_SplitWordsWithLengths {

    public static int getStringLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
            // Exception caught, return count
        }
        return count;
    }

    public static String[] splitUsingCharAt(String text) {
        int length = getStringLength(text);
        int spaceCount = 0;
```

```
    for (int i = 0; i < length; i++) {
        if (text.charAt(i) == ' ') {
            spaceCount++;
        }
    }

    String[] words = new String[spaceCount + 1];
    int wordIndex = 0;
    int start = 0;

    for (int i = 0; i < length; i++) {
        if (text.charAt(i) == ' ') {
            words[wordIndex++] = text.substring(start, i);
            start = i + 1;
        }
    }
    words[wordIndex] = text.substring(start, length);

    return words;
}

public static String[][] getWordsWithLengths(String[] words) {
    String[][] result = new String[words.length][2];
    for (int i = 0; i < words.length; i++) {
        result[i][0] = words[i];
        result[i][1] = String.valueOf(getStringLength(words[i]));
    }
    return result;
}

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

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

    String[] words = splitUsingCharAt(input);
    String[][] wordsWithLengths = getWordsWithLengths(words);

    System.out.println("Word\tLength");
    for (String[] pair : wordsWithLengths) {
        System.out.println(pair[0] + "\t" + Integer.parseInt(pair[1]));
    }
```

```
        scanner.close();
    }
}

--- Question4_ShortestLongestWords.java ---
import java.util.Scanner;

public class Question4_ShortestLongestWords {

    public static int getStringLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
            // Exception caught, return count
        }
        return count;
    }

    public static String[] splitUsingCharAt(String text) {
        int length = getStringLength(text);
        int spaceCount = 0;

        for (int i = 0; i < length; i++) {
            if (text.charAt(i) == ' ') {
                spaceCount++;
            }
        }

        String[] words = new String[spaceCount + 1];
        int wordIndex = 0;
        int start = 0;

        for (int i = 0; i < length; i++) {
            if (text.charAt(i) == ' ') {
                words[wordIndex++] = text.substring(start, i);
                start = i + 1;
            }
        }
        words[wordIndex] = text.substring(start, length);
```

```
    return words;
}

public static String[][] getWordsWithLengths(String[] words) {
    String[][] result = new String[words.length][2];
    for (int i = 0; i < words.length; i++) {
        result[i][0] = words[i];
        result[i][1] = String.valueOf(getStringLength(words[i]));
    }
    return result;
}

public static int[] findShortestLongest(String[][] wordsWithLengths) {
    int shortestIndex = 0;
    int longestIndex = 0;
    int shortestLength = Integer.MAX_VALUE;
    int longestLength = Integer.MIN_VALUE;

    for (int i = 0; i < wordsWithLengths.length; i++) {
        int length = Integer.parseInt(wordsWithLengths[i][1]);
        if (length < shortestLength) {
            shortestLength = length;
            shortestIndex = i;
        }
        if (length > longestLength) {
            longestLength = length;
            longestIndex = i;
        }
    }
    return new int[]{shortestIndex, longestIndex};
}

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

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

    String[] words = splitUsingCharAt(input);
    String[][] wordsWithLengths = getWordsWithLengths(words);
    int[] shortestLongestIndices = findShortestLongest(wordsWithLengths);

    System.out.println("Shortest word: " + wordsWithLengths[shortestLongestIndices[0]][0] +
```

```
            " (Length: " + wordsWithLengths[shortestLongestIndices[0]][1] + ")");
        System.out.println("Longest word: " + wordsWithLengths[shortestLongestIndices[1]][0] +
            " (Length: " + wordsWithLengths[shortestLongestIndices[1]][1] + ")");

        scanner.close();
    }
}

--- Question5_VowelConsonantCount.java ---
import java.util.Scanner;

public class Question5_VowelConsonantCount {

    public static boolean isVowel(char ch) {
        ch = Character.toLowerCase(ch);
        return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
    }

    public static String checkCharType(char ch) {
        if (!Character.isLetter(ch)) {
            return "Not a Letter";
        }
        return isVowel(ch) ? "Vowel" : "Consonant";
    }

    public static int[] countVowelsAndConsonants(String str) {
        int vowels = 0;
        int consonants = 0;
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (Character.isLetter(ch)) {
                if (isVowel(ch)) {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }
        return new int[]{vowels, consonants};
    }

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

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

        int[] counts = countVowelsAndConsonants(input);

        System.out.println("Number of vowels: " + counts[0]);
        System.out.println("Number of consonants: " + counts[1]);

        scanner.close();
    }
}

--- Question6_VowelConsonantCharType.java ---
import java.util.Scanner;

public class Question6_VowelConsonantCharType {

    public static boolean isVowel(char ch) {
        ch = Character.toLowerCase(ch);
        return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
    }

    public static String checkCharType(char ch) {
        if (!Character.isLetter(ch)) {
            return "Not a Letter";
        }
        return isVowel(ch) ? "Vowel" : "Consonant";
    }

    public static String[][] getCharTypes(String str) {
        String[][] result = new String[str.length()][2];
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            result[i][0] = String.valueOf(ch);
            result[i][1] = checkCharType(ch);
        }
        return result;
    }

    public static void displayTable(String[][] data) {
        System.out.println("Character\tType");
        for (String[] row : data) {
            System.out.println(row[0] + "\t\t" + row[1]);
        }
```

```
    }

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

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

        String[][] charTypes = getCharTypes(input);
        displayTable(charTypes);

        scanner.close();
    }
}

--- Question7_TrimSpaces.java ---
import java.util.Scanner;

public class Question7_TrimSpaces {

    public static int[] trimSpaces(String str) {
        int start = 0;
        int end = str.length() - 1;

        // Trim leading spaces
        while (start <= end && str.charAt(start) == ' ') {
            start++;
        }

        // Trim trailing spaces
        while (end >= start && str.charAt(end) == ' ') {
            end--;
        }

        return new int[]{start, end + 1}; // end + 1 for substring end index
    }

    public static String substringUsingCharAt(String str, int start, int end) {
        StringBuilder sb = new StringBuilder();
        for (int i = start; i < end; i++) {
            sb.append(str.charAt(i));
        }
        return sb.toString();
    }
```

```
    public static boolean compareUsingCharAt(String s1, String s2) {
        if (s1.length() != s2.length()) {
            return false;
        }
        for (int i = 0; i < s1.length(); i++) {
            if (s1.charAt(i) != s2.charAt(i)) {
                return false;
            }
        }
        return true;
    }

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

        System.out.print("Enter a string with leading and trailing spaces: ");
        String input = scanner.nextLine();

        int[] trimIndices = trimSpaces(input);
        String trimmedUserDefined = substringUsingCharAt(input, trimIndices[0], trimIndices[1]);
        String trimmedBuiltIn = input.trim();

        System.out.println("Trimmed using charAt(): '" + trimmedUserDefined + "'");
        System.out.println("Trimmed using trim(): '" + trimmedBuiltIn + "'");

        boolean isEqual = compareUsingCharAt(trimmedUserDefined, trimmedBuiltIn);
        System.out.println("Are both trimmed strings equal? " + isEqual);

        scanner.close();
    }
}

--- Question8_VotingEligibility.java ---
import java.util.Random;

public class Question8_VotingEligibility {

    public static int[] generateRandomAges(int n) {
        Random random = new Random();
        int[] ages = new int[n];
        for (int i = 0; i < n; i++) {
            ages[i] = random.nextInt(90) + 10; // random 2-digit age between 10 and 99
        }
```

```
        return ages;
    }

    public static String[][] checkVotingEligibility(int[] ages) {
        String[][] result = new String[ages.length][2];
        for (int i = 0; i < ages.length; i++) {
            result[i][0] = String.valueOf(ages[i]);
            if (ages[i] < 0) {
                result[i][1] = "Cannot Vote";
            } else if (ages[i] >= 18) {
                result[i][1] = "Can Vote";
            } else {
                result[i][1] = "Cannot Vote";
            }
        }
        return result;
    }

    public static void displayResults(String[][] data) {
        System.out.println("Age\tEligibility");
        for (String[] row : data) {
            System.out.println(row[0] + "\t" + row[1]);
        }
    }

    public static void main(String[] args) {
        int n = 10; // number of students
        int[] ages = generateRandomAges(n);
        String[][] eligibility = checkVotingEligibility(ages);
        displayResults(eligibility);
    }
}

--- Question9_RockPaperScissors.java ---
import java.util.Scanner;

public class Question9_RockPaperScissors {

    public static String getComputerChoice() {
        double rand = Math.random();
        if (rand < 1.0 / 3) {
            return "rock";
        } else if (rand < 2.0 / 3) {
            return "paper";
```

```
        } else {
            return "scissors";
        }
    }

    public static String findWinner(String userChoice, String computerChoice) {
        if (userChoice.equals(computerChoice)) {
            return "Draw";
        }
        if (userChoice.equals("rock")) {
            return computerChoice.equals("scissors") ? "User" : "Computer";
        } else if (userChoice.equals("paper")) {
            return computerChoice.equals("rock") ? "User" : "Computer";
        } else if (userChoice.equals("scissors")) {
            return computerChoice.equals("paper") ? "User" : "Computer";
        }
        return "Invalid";
    }

    public static String[][] calculateStats(int userWins, int computerWins, int draws, int
totalGames) {
        String[][] stats = new String[4][2];
        stats[0][0] = "User Wins";
        stats[0][1] = String.valueOf(userWins);
        stats[1][0] = "Computer Wins";
        stats[1][1] = String.valueOf(computerWins);
        stats[2][0] = "Draws";
        stats[2][1] = String.valueOf(draws);
        stats[3][0] = "Total Games";
        stats[3][1] = String.valueOf(totalGames);
        return stats;
    }

    public static void displayStats(String[][] stats) {
        System.out.println("Stat\tValue");
        for (String[] stat : stats) {
            System.out.println(stat[0] + "\t" + stat[1]);
        }
    }

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

        System.out.print("Enter number of games: ");
```

```
        int numGames = scanner.nextInt();
        scanner.nextLine(); // consume newline

        int userWins = 0;
        int computerWins = 0;
        int draws = 0;

        for (int i = 1; i <= numGames; i++) {
            System.out.print("Game " + i + " - Enter your choice (rock, paper, scissors): ");
            String userChoice = scanner.nextLine().toLowerCase();

            String computerChoice = getComputerChoice();
            System.out.println("Computer choice: " + computerChoice);

            String winner = findWinner(userChoice, computerChoice);
            if (winner.equals("User")) {
                userWins++;
                System.out.println("You win!");
            } else if (winner.equals("Computer")) {
                computerWins++;
                System.out.println("Computer wins!");
            } else if (winner.equals("Draw")) {
                draws++;
                System.out.println("It's a draw!");
            } else {
                System.out.println("Invalid input. No winner for this game.");
            }
        }

        String[][] stats = calculateStats(userWins, computerWins, draws, numGames);
        displayStats(stats);

        double userWinPercent = (userWins * 100.0) / numGames;
        double computerWinPercent = (computerWins * 100.0) / numGames;

        System.out.printf("User winning percentage: %.2f%%\n", userWinPercent);
        System.out.printf("Computer winning percentage: %.2f%%\n", computerWinPercent);

        scanner.close();
    }
}

--- Question10_StudentGrades.java ---
import java.util.Random;
```

```
public class Question10_StudentGrades {

    public static int[][] generateRandomScores(int n) {
        Random random = new Random();
        int[][] scores = new int[n][3];
        for (int i = 0; i < n; i++) {
            scores[i][0] = random.nextInt(100); // Physics
            scores[i][1] = random.nextInt(100); // Chemistry
            scores[i][2] = random.nextInt(100); // Maths
        }
        return scores;
    }

    public static double roundToTwoDecimals(double value) {
        return Math.round(value * 100.0) / 100.0;
    }

    public static double[][] calculateTotalsAndPercentages(int[][] scores) {
        int n = scores.length;
        double[][] results = new double[n][4]; // total, average, percentage, placeholder for grade
index
        for (int i = 0; i < n; i++) {
            int total = scores[i][0] + scores[i][1] + scores[i][2];
            double average = total / 3.0;
            double percentage = (total / 300.0) * 100.0;
            results[i][0] = total;
            results[i][1] = roundToTwoDecimals(average);
            results[i][2] = roundToTwoDecimals(percentage);
        }
        return results;
    }

    public static String[] calculateGrades(double[][] results) {
        String[] grades = new String[results.length];
        for (int i = 0; i < results.length; i++) {
            double percentage = results[i][2];
            if (percentage >= 90) {
                grades[i] = "A+";
            } else if (percentage >= 80) {
                grades[i] = "A";
            } else if (percentage >= 70) {
                grades[i] = "B+";
            } else if (percentage >= 60) {
```

```
            grades[i] = "B";
        } else if (percentage >= 50) {
            grades[i] = "C";
        } else if (percentage >= 40) {
            grades[i] = "D";
        } else {
            grades[i] = "F";
        }
    }
    return grades;
}

public static void displayScorecard(int[][] scores, double[][] results, String[] grades) {

System.out.println("Student\tPhysics\tChemistry\tMaths\tTotal\tAverage\tPercentage\tGrade");
    for (int i = 0; i < scores.length; i++) {
        System.out.printf("%d\t%d\t%d\t\t%d\t%d\t%.2f\t%.2f\t\t%s\n",
                i + 1,
                scores[i][0],
                scores[i][1],
                scores[i][2],
                (int) results[i][0],
                results[i][1],
                results[i][2],
                grades[i]);
    }
}

public static void main(String[] args) {
    int n = 10; // number of students
    int[][] scores = generateRandomScores(n);
    double[][] results = calculateTotalsAndPercentages(scores);
    String[] grades = calculateGrades(results);
    displayScorecard(scores, results, grades);
}
}
```