Level 2 Practice Programs

1. **Write a program to find and return the length of a string without using the *length()* method.**

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

    //Finding Length of the string
    public static int findLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (StringIndexOutOfBoundsException e) {
        }
        return count;
    }

    //User defined method and built in function
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = input.next();
        int customLength = findLength(str);
        int builtInLength = str.length();
        System.out.println("Length (using custom method): " + customLength);
        System.out.println("Length (using built-in method): " + builtInLength);
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac StringLengthFinder.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java StringLengthFinder
Enter a string: Shounak
Length (using custom method): 7
Length (using built-in method): 7
```

2. **Write a program to split the text into words, compare the result with the split() method and display the result.**

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

    // Method to find length of a string without using length()
    public static int findLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (Exception e) {
        }
        return count;
    }

    // Method to split a string into words without using split()
    public static String[] customSplit(String str) {
        int len = findLength(str);
        int wordCount = 1;

        // Count spaces to determine number of words
        for (int i = 0; i < len; i++) {
            if (str.charAt(i) == ' ') {
                wordCount++;
            }
        }

        String[] words = new String[wordCount];
        String word = "";
        int index = 0;

        // Build words by checking spaces
        for (int i = 0; i < len; i++) {
            char ch = str.charAt(i);
            if (ch != ' ') {
                word += ch;
            } else {
                words[index++] = word;
                word = "";
            }
        }
        words[index] = word;

        return words;
    }
```

```java
    // Method to compare two string arrays
    public static boolean compareArrays(String[] a, String[] b) {
        if (a.length != b.length) return false;
        for (int i = 0; i < a.length; i++) {
            if (!a[i].equals(b[i])) return false;
        }
        return true;
    }

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

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

        String[] custom = customSplit(text);
        String[] builtin = text.split(" ");
        boolean same = compareArrays(custom, builtin);

        System.out.println("Custom Split:");
        for (String w : custom) System.out.println(w);

        System.out.println("Built-in Split:");
        for (String w : builtin) System.out.println(w);

        System.out.println("Are both results equal? " + same);
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac SimpleStringSplitter.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java SimpleStringSplitter
Enter a sentence: My name is Shounak
Custom Split:
My
name
is
Shounak
Built-in Split:
My
name
is
Shounak
Are both results equal? true
```

3. **Write a program to split the text into words and return the words along with their lengths in a 2D array.**

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

    // Method to find string length without using length()
    public static int getLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (Exception e) {
        }
        return count;
    }

    // Method to split text into words without using split()
    public static String[] splitWords(String text) {
        int len = getLength(text);
        int wordCount = 1;

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

        String[] words = new String[wordCount];
        String word = "";
        int index = 0;

        for (int i = 0; i < len; i++) {
            char ch = text.charAt(i);
            if (ch != ' ') {
                word += ch;
            } else {
                words[index++] = word;
                word = "";
            }
        }
        words[index] = word;
        return words;
    }
```

```java
    // Method to return a 2D String array with word and its length
    public static String[][] getWordLengthPairs(String[] words) {
        String[][] wordInfo = new String[words.length][2];
        for (int i = 0; i < words.length; i++) {
            wordInfo[i][0] = words[i];
            wordInfo[i][1] = String.valueOf(getLength(words[i]));
        }
        return wordInfo;
    }

    // Main method
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Take input
        System.out.print("Enter a sentence: ");
        String line = input.nextLine();

        // Get words and their lengths
        String[] words = splitWords(line);
        String[][] wordData = getWordLengthPairs(words);

        // Display table
        System.out.printf("%-15s %-10s\n", "Word", "Length");
        for (int i = 0; i < wordData.length; i++) {
            String word = wordData[i][0];
            int length = Integer.parseInt(wordData[i][1]); // convert from String to int
            System.out.printf("%-15s %-10d\n", word, length);
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac WordLengthTable.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java WordLengthTable
Enter a sentence: My name is Shounak
Word            Length
My              2
name            4
is              2
Shounak         7
```

4. **Write a program to split the text into words and find the shortest and longest strings in a given text.**

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

    public static int getLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (Exception e) {
        }
        return count;
    }

    public static String[] splitWords(String text) {
        int len = getLength(text);
        int wordCount = 1;

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

        String[] words = new String[wordCount];
        String word = "";
        int index = 0;

        for (int i = 0; i < len; i++) {
            char ch = text.charAt(i);
            if (ch != ' ') {
                word += ch;
            } else {
                words[index++] = word;
                word = "";
            }
        }
        words[index] = word;
        return words;
    }
```

```
    public static String[][] getWordLengthPairs(String[] words) {
        String[][] wordInfo = new String[words.length][2];
        for (int i = 0; i < words.length; i++) {
            wordInfo[i][0] = words[i];
            wordInfo[i][1] = String.valueOf(getLength(words[i]));
        }
        return wordInfo;
    }

    public static int[] findShortestAndLongest(String[][] wordInfo) {
        int minLen = Integer.parseInt(wordInfo[0][1]);
        int maxLen = Integer.parseInt(wordInfo[0][1]);
        int minIndex = 0;
        int maxIndex = 0;

        for (int i = 1; i < wordInfo.length; i++) {
            int len = Integer.parseInt(wordInfo[i][1]);
            if (len < minLen) {
                minLen = len;
                minIndex = i;
            }
            if (len > maxLen) {
                maxLen = len;
                maxIndex = i;
            }
        }

        return new int[]{minIndex, maxIndex};
    }

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

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

        String[] words = splitWords(line);
        String[][] wordInfo = getWordLengthPairs(words);
        int[] result = findShortestAndLongest(wordInfo);

        System.out.println("\nWord\t\tLength");
        for (int i = 0; i < wordInfo.length; i++) {
            System.out.println(wordInfo[i][0] + "\t\t" + Integer.parseInt(wordInfo[i][1]));
        }
```

```
        System.out.println("\nShortest word: " + wordInfo[result[0]][0]);
        System.out.println("Longest word: " + wordInfo[result[1]][0]);
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac ShortestLongestWordFinder.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java ShortestLongestWordFinder
Enter a sentence: My name is Shounak

Word            Length
My              2
name            4
is              2
Shounak         7

Shortest word: My
Longest word: Shounak
```

5. **Write a program to find vowels and consonants in a string and display the count of Vowels and Consonants in the string.**

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

    // Method to check if a character is a vowel, consonant, or not a letter
    public static String checkCharType(char ch) {
        // Convert to lowercase if it's an uppercase letter
        if (ch >= 'A' && ch <= 'Z') {
            ch = (char)(ch + 32); // ASCII conversion to lowercase
        }

        // Check if it's a letter
        if (ch >= 'a' && ch <= 'z') {
            // Check if it's a vowel
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                return "Vowel";
            } else {
                return "Consonant";
            }
        } else {
            return "Not a Letter";
        }
    }

    // Method to count vowels and consonants in a string
    public static int[] countVowelsAndConsonants(String str) {
        int vowelCount = 0;
        int consonantCount = 0;

        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            String charType = checkCharType(currentChar);

            if (charType.equals("Vowel")) {
                vowelCount++;
            } else if (charType.equals("Consonant")) {
                consonantCount++;
            }
            // Ignore non-letter characters
        }

        return new int[]{vowelCount, consonantCount};
    }
```

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

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

        int[] counts = countVowelsAndConsonants(inputString);

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

        scanner.close();
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac VowelConsonantCounter.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java VowelConsonantCounter
Enter a string: Shounak Roy
Number of vowels: 4
Number of consonants: 6
```

6. **Write a program to find vowels and consonants in a string and display the character type - Vowel, Consonant, or Not a Letter.**

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

    // Method to check if a character is a vowel, consonant, or not a letter
    public static String checkCharType(char ch) {
        // Convert to lowercase if it's an uppercase letter
        if (ch >= 'A' && ch <= 'Z') {
            ch = (char)(ch + 32); // ASCII conversion to lowercase
        }

        // Check if it's a letter
        if (ch >= 'a' && ch <= 'z') {
            // Check if it's a vowel
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                return "Vowel";
            } else {
                return "Consonant";
            }
        } else {
            return "Not a Letter";
        }
    }

    // Method to analyze each character in the string
    public static String[][] analyzeString(String str) {
        String[][] result = new String[str.length()][2];

        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            result[i][0] = String.valueOf(currentChar);
            result[i][1] = checkCharType(currentChar);
        }

        return result;
    }
```

```java
    // Method to display the 2D array in a tabular format
    public static void displayResults(String[][] charAnalysis) {
        System.out.println("\nCharacter Analysis:");
        System.out.println("+------------+----------------+");
        System.out.println("| Character  |      Type      |");
        System.out.println("+------------+----------------+");

        for (String[] row : charAnalysis) {
            System.out.printf("|     %-6s | %-14s |\n", row[0], row[1]);
        }

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

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

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

        String[][] analysis = analyzeString(inputString);
        displayResults(analysis);

        scanner.close();
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac VowelConsonantIdentifier.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java VowelConsonantIdentifier
Enter a string: Shounak

Character Analysis:
+------------+----------------+
| Character  |      Type      |
+------------+----------------+
|     S      | Consonant      |
|     h      | Consonant      |
|     o      | Vowel          |
|     u      | Vowel          |
|     n      | Consonant      |
|     a      | Vowel          |
|     k      | Consonant      |
+------------+----------------+
```

7. **Write a program to trim the leading and trailing spaces from a string using the *charAt()* method.**

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

    // Method to find the start and end indices of non-space characters
    public static int[] findTrimIndices(String str) {
        int start = 0;
        int end = str.length() - 1;

        // Find first non-space character from start
        while (start <= end && str.charAt(start) == ' ') {
            start++;
        }

        // Find first non-space character from end
        while (end >= start && str.charAt(end) == ' ') {
            end--;
        }

        return new int[]{start, end};
    }

    // Method to create substring using charAt()
    public static String customSubstring(String str, int start, int end) {
        // Handle empty string case
        if (start > end) {
            return "";
        }

        StringBuilder result = new StringBuilder();
        for (int i = start; i <= end; i++) {
            result.append(str.charAt(i));
        }

        return result.toString();
    }
```

```java
    // Method to compare two strings using charAt()
    public static boolean compareStrings(String str1, String str2) {
        if (str1.length() != str2.length()) {
            return false;
        }

        for (int i = 0; i < str1.length(); i++) {
            if (str1.charAt(i) != str2.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/trailing spaces: ");
        String inputString = scanner.nextLine();

        // Custom trim using charAt()
        int[] indices = findTrimIndices(inputString);
        String customTrimmed = customSubstring(inputString, indices[0], indices[1]);

        // Built-in trim for comparison
        String builtInTrimmed = inputString.trim();

        // Compare results
        boolean areEqual = compareStrings(customTrimmed, builtInTrimmed);

        System.out.println("\nOriginal string: '" + inputString + "'");
        System.out.println("Custom trimmed: '" + customTrimmed + "'");
        System.out.println("Built-in trimmed: '" + builtInTrimmed + "'");
        System.out.println("Are they equal? " + areEqual);

        scanner.close();
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac CustomStringTrimmer.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java CustomStringTrimmer
Enter a string with leading/trailing spaces: Shounak Roy

Original string: 'Shounak Roy'
Custom trimmed: 'Shounak Roy'
Built-in trimmed: 'Shounak Roy'
Are they equal? true
```

8. **Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.**

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

    // Method to generate random 2-digit ages for n students
    public static int[] generateStudentAges(int numStudents) {
        int[] ages = new int[numStudents];
        for (int i = 0; i < numStudents; i++) {
            ages[i] = (int)(Math.random() * 50) + 10; // Ages between 10-
        }
        return ages;
    }

    // Method to check voting eligibility
    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] = "Invalid (Negative)";
            } else if (ages[i] >= 18) {
                result[i][1] = "Can Vote";
            } else {
                result[i][1] = "Cannot Vote";
            }
        }

        return result;
    }

    // Method to display results in tabular format
    public static void displayResults(String[][] eligibilityData) {
        System.out.println("\nStudent Voting Eligibility Report:");
        System.out.println("+------------+-------------------+");
        System.out.println("| Student Age| Voting Eligibility |");
        System.out.println("+------------+-------------------+");

        for (String[] row : eligibilityData) {
            System.out.printf("| %10s | %17s |\n", row[0], row[1]);
        }

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

```java
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        final int NUM_STUDENTS = 10;

        System.out.println("Enter ages for " + NUM_STUDENTS + " students:");

        // Option 1: Manual input
        int[] studentAges = new int[NUM_STUDENTS];
        for (int i = 0; i < NUM_STUDENTS; i++) {
            System.out.print("Enter age for student " + (i+1) + ": ");
            studentAges[i] = scanner.nextInt();
        }

        // Option 2: Random generation (uncomment to use instead of manual input)
        // int[] studentAges = generateStudentAges(NUM_STUDENTS);

        String[][] eligibility = checkVotingEligibility(studentAges);
        displayResults(eligibility);

        scanner.close();
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac VotingEligibilityChecker.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java VotingEligibilityChecker
Enter ages for 10 students:
Enter age for student 1: 18
Enter age for student 2: 16
Enter age for student 3: 22
Enter age for student 4: 32
Enter age for student 5: 14
Enter age for student 6: 19
Enter age for student 7: 19
Enter age for student 8: 20
Enter age for student 9: 15
Enter age for student 10: 24

Student Voting Eligibility Report:
+------------+-------------------+
| Student Age| Voting Eligibility |
+------------+-------------------+
|         18 |          Can Vote |
|         16 |       Cannot Vote |
|         22 |          Can Vote |
|         32 |          Can Vote |
|         14 |       Cannot Vote |
|         19 |          Can Vote |
|         19 |          Can Vote |
|         20 |          Can Vote |
|         15 |       Cannot Vote |
|         24 |          Can Vote |
+------------+-------------------+
```

9. **Rock-Paper-Scissors is a game played between a minimum of two players. Each player can choose either rock, paper, or scissors. Here the game is played between a user and a computer. Based on the rules, either a player**

or a computer will win. Show the stats of player and computer win in a tabular format across multiple games. Also, show the winning percentage between the player and the computer..

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

    // Method to get computer's choice using Math.random()
    public static String getComputerChoice() {
        int randomNum = (int)(Math.random() * 3);
        switch(randomNum) {
            case 0: return "rock";
            case 1: return "paper";
            case 2: return "scissors";
            default: return "rock"; // fallback
        }
    }

    // Method to determine the winner
    public static String determineWinner(String userChoice, String computerChoice) {
        if (userChoice.equals(computerChoice)) {
            return "draw";
        }

        if ((userChoice.equals("rock") && computerChoice.equals("scissors")) ||
            (userChoice.equals("paper") && computerChoice.equals("rock")) ||
            (userChoice.equals("scissors") && computerChoice.equals("paper"))) {
            return "user";
        } else {
            return "computer";
        }
    }

    // Method to calculate statistics
    public static String[][] calculateStats(int[] results, int totalGames) {
        int userWins = results[0];
        int computerWins = results[1];
        int draws = results[2];

        double userWinPercent = (double)userWins / totalGames * 100;
        double computerWinPercent = (double)computerWins / totalGames * 100;
        double drawPercent = (double)draws / totalGames * 100;

        return new String[][] {
            {"User Wins", String.valueOf(userWins), String.format("%.2f%%", userWinPercent)},
            {"Computer Wins", String.valueOf(computerWins), String.format("%.2f%%", computerWinPercent)},
            {"Draws", String.valueOf(draws), String.format("%.2f%%", drawPercent)},
            {"Total Games", String.valueOf(totalGames), "100.00%"}
        };
    }
```

```java
    // Method to display results
    public static void displayResults(String[][] gameHistory, String[][] stats) {
        System.out.println("\nGame History:");
        System.out.println("+---------+------------+-----------------+--------+");
        System.out.println("| Game #  | Your Choice| Computer Choice | Result |");
        System.out.println("+---------+------------+-----------------+--------+");

        for (int i = 0; i < gameHistory.length; i++) {
            System.out.printf("| %7d | %10s | %15s | %6s |\n",
                              (i+1), gameHistory[i][0], gameHistory[i][1], gameHistory[i][2]);
        }
        System.out.println("+---------+------------+-----------------+--------+");

        System.out.println("\nGame Statistics:");
        System.out.println("+----------------+------------+------------+");
        System.out.println("| Category       | Count      | Percentage |");
        System.out.println("+----------------+------------+------------+");

        for (String[] stat : stats) {
            System.out.printf("| %-14s | %-10s | %-10s |\n", stat[0], stat[1], stat[2]);
        }
        System.out.println("+----------------+------------+------------+");
    }
```

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

        System.out.print("How many games would you like to play? ");
        int numGames = scanner.nextInt();

        String[][] gameHistory = new String[numGames][3];
        int[] results = new int[3]; // [0]=user wins, [1]=computer wins, [2]=draws

        for (int i = 0; i < numGames; i++) {
            System.out.print("\nGame " + (i+1) + ": Enter your choice (rock/paper/scissors): ");
            String userChoice = scanner.next().toLowerCase();

            // Validate input
            while (!userChoice.equals("rock") && !userChoice.equals("paper") && !userChoice.equals("scissors")) {
                System.out.print("Invalid choice. Please enter rock, paper, or scissors: ");
                userChoice = scanner.next().toLowerCase();
            }

            String computerChoice = getComputerChoice();
            String result = determineWinner(userChoice, computerChoice);

            // Record game history
            gameHistory[i][0] = userChoice;
            gameHistory[i][1] = computerChoice;
            gameHistory[i][2] = result.equals("draw") ? "Draw" :
                                (result.equals("user") ? "You Win" : "Computer Wins");

            // Update results
            if (result.equals("user")) {
                results[0]++;
            } else if (result.equals("computer")) {
                results[1]++;
            } else {
                results[2]++;
            }
        }

        String[][] stats = calculateStats(results, numGames);
        displayResults(gameHistory, stats);

        scanner.close();
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac RockPaperScissorsGame.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java RockPaperScissorsGame
How many games would you like to play? 2

Game 1: Enter your choice (rock/paper/scissors): rock

Game 2: Enter your choice (rock/paper/scissors): papaer
Invalid choice. Please enter rock, paper, or scissors: paper

Game History:
+---------+------------+-----------------+--------+
| Game #  | Your Choice| Computer Choice | Result |
+---------+------------+-----------------+--------+
|       1 |       rock |        scissors | You Win |
|       2 |      paper |           paper |   Draw |
+---------+------------+-----------------+--------+

Game Statistics:
+---------------+------------+------------+
| Category      | Count      | Percentage |
+---------------+------------+------------+
| User Wins     | 1          | 50.00%     |
| Computer Wins | 0          | 0.00%      |
| Draws         | 1          | 50.00%     |
| Total Games   | 2          | 100.00%    |
+---------------+------------+------------+
```

10. **Create a program to take input marks of students in 3 subjects physics, chemistry, and maths. Compute the percentage and then calculate the grade as shown in figure below.**

```java
import java.util.Scanner;
import java.util.Random;
public class StudentScorecard {

    // Generate random 2-digit marks for Physics, Chemistry, and Math
    public static int[][] generateScores(int students) {
        Random rand = new Random();
        int[][] scores = new int[students][3];
        for (int i = 0; i < students; i++) {
            for (int j = 0; j < 3; j++) {
                scores[i][j] = rand.nextInt(100); // Random marks 0-99
            }
        }
        return scores;
    }

    // Calculate total, average, and percentage
    public static double[][] calculateResults(int[][] scores) {
        int students = scores.length;
        double[][] results = new double[students][3]; // total, average, percentage
        for (int i = 0; i < students; i++) {
            int total = scores[i][0] + scores[i][1] + scores[i][2];
            double average = total / 3.0;
            double percentage = (total / 300.0) * 100;
            results[i][0] = total;
            results[i][1] = Math.round(average * 100.0) / 100.0;
            results[i][2] = Math.round(percentage * 100.0) / 100.0;
        }
        return results;
    }

    // Determine grade from percentage
    public static char[] calculateGrades(double[][] results) {
        int students = results.length;
        char[] grades = new char[students];
        for (int i = 0; i < students; i++) {
            double percent = results[i][2];
            if (percent >= 80) grades[i] = 'A';
            else if (percent >= 70) grades[i] = 'B';
            else if (percent >= 60) grades[i] = 'C';
            else if (percent >= 50) grades[i] = 'D';
            else if (percent >= 40) grades[i] = 'E';
            else grades[i] = 'R';
        }
        return grades;
    }
```

```java
    // Display final scorecard in table format
    public static void displayScorecard(int[][] scores, double[][] results, char[] grades) {
        System.out.printf("%-10s %-10s %-10s %-10s %-10s %-10s %-10s %-10s\n",
                "Student", "Physics", "Chemistry", "Math", "Total", "Average", "Percent", "Grade");

        for (int i = 0; i < scores.length; i++) {
            System.out.printf("%-10d %-10d %-10d %-10d %-10.0f %-10.2f %-10.2f %-10c\n",
                    i + 1, scores[i][0], scores[i][1], scores[i][2],
                    results[i][0], results[i][1], results[i][2], grades[i]);
        }
    }

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

        // Take number of students from user
        System.out.print("Enter number of students: ");
        int n = input.nextInt();

        // Generate data and process
        int[][] scores = generateScores(n);
        double[][] results = calculateResults(scores);
        char[] grades = calculateGrades(results);

        // Display scorecard
        displayScorecard(scores, results, grades);
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>javac StudentScorecard.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 5 - Strings\Level 2>java StudentScorecard
Enter number of students: 5
Student    Physics    Chemistry  Math       Total      Average    Percent    Grade
1          45         20         80         145        48.33      48.33      E
2          95         80         22         197        65.67      65.67      C
3          69         63         93         225        75.00      75.00      B
4          57         98         37         192        64.00      64.00      C
5          41         5          3          49         16.33      16.33      R
```