CodInClub

Powered By BridgeLabz

Best Programming Practice

1. Use Variables including for Fixed, User Inputs, and Results
2. Use Methods instead of writing code in the main() function
3. Proper naming conventions for all variables and methods
4. Proper Program Name and Class Name
5. Handle Checked and Unchecked Exceptions wherever possible
6. Proper Method Name which indicates action taking inputs and providing result

Sample Program 1: Create a program to find all the occurrences of a character in a string using charAt() method

a. Take user input for the String and occurrences of the Character to find
b. Write a method to find all the occurrences of the characters.
    i. The logic used is to first find the number of occurrences of the character and
    ii. then create an array to store the indexes of the character
c. Call the method in the main and display the result

Java

```java
// Program to find all the occurrences of a character in a string
import java.util.Scanner;
class StringAnalyzer {
  // Method to find all the index of a character in a string using charAt()
  // method and return them in an array
  public static int[] findAllIndexes(String text, char ch) {
    // The count is used to find the number of occurrences of the character
    int count = 0;
    for (int i = 0; i < text.length(); i++) {
        if (text.charAt(i) == ch) {
            count++;
        }
    }

    // Create an array to store the indexes of the character
    int[] indexes = new int[count];
    int j = 0;
    for (int i = 0; i < text.length(); i++) {
        if (text.charAt(i) == ch) {
            indexes[j] = i;
            j++;
        }
    }
    return indexes;
```

```
  }
  public static void main(String[] args) {
    // Take user input for Text and Character to check Occurrences
    Scanner sc = new Scanner(System.in);
    System.out.print(Enter a text: ");
    String text = sc.nextLine();
    System.out.print("Enter a character to find the occurrences: ");
    char ch = sc.next().charAt(0);

    // Find the occurrences of the character
    int[] indexes = findAllIndexes(text, ch);

    // Display the occurrences of the character
    System.out.println("Indexes of the character '" + ch + "': ");
    for (int i = 0; i < indexes.length; i++) {
        System.out.print(indexes[i] + " ");
    }
  }
}
```

Level 2 Practice Programs

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

   Hint =>

   a. Take user input using the ***Scanner next()*** method
   b. Create a method to find and return a string's length without using the built-in length() method. The logic for this is to use the infinite loop to count each character till the ***charAt()*** method throws a runtime exception, handles the exception, and then return the count
   c. The main function calls the user-defined method as well as the built-in ***length()*** method and displays the result

```
package Day5.LabPractice_L2;

import java.util.Scanner;

public class LP1 {
   public static int findStringLength(String text) {
        int count = 0;
        try {
            while (true) {
                text.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
        }
        return count;
   }
   public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputText = scanner.next();
        int builtInLength = inputText.length();
        int customLength = findStringLength(inputText);
        System.out.println("\nOriginal Text: " + inputText);
        System.out.println("Length using built-in method: " +
builtInLength);
```

```
        System.out.println("Length using custom method: " + customLength);
        System.out.println("Are both lengths equal? " + (builtInLength ==
customLength));


        scanner.close();
    }
}
```

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

   Hint =>

   a. Take user input using the ***Scanner*** ***nextLine()*** method
   b. Create a Method to find the length of the String without using the built-in length() method.
   c. Create a Method to split the text into words using the charAt() method without using the String built-in ***split()*** method and return the words. Use the following logic
      i. Firstly Count the number of words in the text and create an array to store the indexes of the spaces for each word in a 1D array
      ii. Then Create an array to store the words and use the indexes to extract the words
   d. Create a method to compare the two String arrays and return a boolean
   e. The main function calls the user-defined method and the built-in ***split()*** method. Call the user defined method to compare the two string arrays and display the result

```
package Day5.LabPractice_L2;
import java.util.Scanner;
import java.util.Arrays;
public class LP2 {
    public static int findStringLength(String text) {
        int count = 0;
        try {
            while (true) {
                text.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
        }
        return count;
```

```
    }
    public static String[] customSplit(String text) {
        int length = findStringLength(text);
        int wordCount = 1;
        for (int i = 0; i < length; i++) {
            if (text.charAt(i) == ' ') {
                wordCount++;
            }
        }
        String[] words = new String[wordCount];
        int wordIndex = 0, start = 0;
        for (int i = 0; i < length; i++) {
            if (text.charAt(i) == ' ') {
                words[wordIndex++] = extractSubstring(text, start, i);
                start = i + 1;
            }
        }
        words[wordIndex] = extractSubstring(text, start, length);
        return words;
    }
    public static String extractSubstring(String text, int start, int end)
{
        StringBuilder subStr = new StringBuilder();
        for (int i = start; i < end; i++) {
            subStr.append(text.charAt(i));
        }
        return subStr.toString();
    }
    public static boolean compareArrays(String[] arr1, String[] arr2) {
        return Arrays.equals(arr1, arr2);
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Enter a sentence: ");
        String inputText = input.nextLine();
        String[] customWords = customSplit(inputText);
        String[] builtInWords = inputText.split(" ");
        boolean areEqual = compareArrays(customWords, builtInWords);
        System.out.println("\nCustom Split Method Output: " +
Arrays.toString(customWords));
        System.out.println("Built-in split() Method Output: " +
Arrays.toString(builtInWords));
        System.out.println("Are both methods producing the same result? "
+ areEqual);
        input.close();
    }
}
```

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

   Hint =>

   a. Take user input using the ***Scanner*** ***nextLine()*** method
   b. Create a Method to split the text into words using the charAt() method without using the String built-in ***split()*** method and return the words.
   c. Create a method to find and return a string's length without using the length() method.
   d. Create a method to take the word array and return a 2D String array of the word and its corresponding length. Use String built-in function String.valueOf() to generate the String value for the number
   e. The main function calls the user-defined method and displays the result in a tabular format. During display make sure to convert the length value from String to Integer and then display

```
package Day5.LabPractice_L2;
import java.util.Scanner;
public class LP3 {
    public static int findStringLength(String text) {
        int count = 0;
        try {
            while (true) {
                text.charAt(count);
```

```
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
        }
        return count;
    }
    public static String[] customSplit(String text) {
        int length = findStringLength(text);
        int wordCount = 1;
        for (int i = 0; i < length; i++) {
            if (text.charAt(i) == ' ') {
                wordCount++;
            }
        }
        String[] words = new String[wordCount];
        int wordIndex = 0, start = 0;
        for (int i = 0; i < length; i++) {
            if (text.charAt(i) == ' ') {
                words[wordIndex++] = extractSubstring(text, start, i);
                start = i + 1;
            }
        }
        words[wordIndex] = extractSubstring(text, start, length);
        return words;
    }
    public static String extractSubstring(String text, int start, int end)
{
        StringBuilder subStr = new StringBuilder();
        for (int i = start; i < end; i++) {
            subStr.append(text.charAt(i));
        }
        return subStr.toString();
    }
```

```
    public static String[][] getWordsWithLengths(String[] words) {
        int numWords = words.length;
        String[][] wordLengths = new String[numWords][2];
        for (int i = 0; i < numWords; i++) {
            wordLengths[i][0] = words[i];
            wordLengths[i][1] =
String.valueOf(findStringLength(words[i]));
        }
        return wordLengths;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String inputText = scanner.nextLine();
        String[] words = customSplit(inputText);
        String[][] wordsWithLengths = getWordsWithLengths(words);
        System.out.println("\nWord\t\tLength");
        for (String[] row : wordsWithLengths) {
            System.out.println(row[0] + "\t\t" +
Integer.parseInt(row[1]));
        }
        scanner.close();
    }
}
```

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

   Hint =>

   a. Take user input using the ***Scanner nextLine()*** method
   b. Create a Method to split the text into words using the charAt() method without using the String built-in ***split()*** method and return the words.
   c. Create a method to find and return a string's length without using the length() method.
   d. Create a method to take the word array and return a 2D String array of the word and its corresponding length. Use String built-in function String.valueOf() to generate the String value for the number

e. Create a Method that takes the 2D array of word and corresponding length as parameters, find the shortest and longest string and return them in an 1D int array.
f. The main function calls the user-defined methods and displays the result.

```
package Day5.LabPractice_L2;
import java.util.Scanner;
public class LP4 {
   public static int findStringLength(String text) {
       int count = 0;
       try {
           while (true) {
               text.charAt(count);
               count++;
           }
       } catch (IndexOutOfBoundsException e) {
       }
       return count;
   }
   public static String[] customSplit(String text) {
       int length = findStringLength(text);
       int wordCount = 1;
       for (int i = 0; i < length; i++) {
           if (text.charAt(i) == ' ') {
               wordCount++;
           }
       }
       String[] words = new String[wordCount];
       int wordIndex = 0, start = 0;
       for (int i = 0; i < length; i++) {
           if (text.charAt(i) == ' ') {
               words[wordIndex++] = extractSubstring(text, start, i);
               start = i + 1;
           }
       }
```

```
        words[wordIndex] = extractSubstring(text, start, length);
        return words;
    }
    public static String extractSubstring(String text, int start, int end)
{
        StringBuilder subStr = new StringBuilder();
        for (int i = start; i < end; i++) {
            subStr.append(text.charAt(i));
        }
        return subStr.toString();
    }
    public static String[][] getWordsWithLengths(String[] words) {
        int numWords = words.length;
        String[][] wordLengths = new String[numWords][2];
        for (int i = 0; i < numWords; i++) {
            if (words[i] != null) {
                wordLengths[i][0] = words[i];
                wordLengths[i][1] =
String.valueOf(findStringLength(words[i]));
            }
        }
        return wordLengths;
    }
    public static String[] findShortestAndLongest(String[][]
wordsWithLengths) {
        String shortestWord = wordsWithLengths[0][0];
        String longestWord = wordsWithLengths[0][0];

        for (String[] row : wordsWithLengths) {
            if (findStringLength(row[0]) < findStringLength(shortestWord))
{
                shortestWord = row[0];
            }
```

```
            if (findStringLength(row[0]) > findStringLength(longestWord))
{
                longestWord = row[0];
            }
        }
        return new String[]{shortestWord, longestWord};
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String inputText = scanner.nextLine();
        String[] words = customSplit(inputText);
        String[][] wordsWithLengths = getWordsWithLengths(words);
        String[] result = findShortestAndLongest(wordsWithLengths);
        System.out.println("\nWord\t\tLength");
        for (String[] row : wordsWithLengths) {
            System.out.println(row[0] + "\t\t" + row[1]);
        }
        System.out.println("\nShortest Word: " + result[0]);
        System.out.println("Longest Word: " + result[1]);
        scanner.close();
    }
}
```

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

   Hint =>

   a. Create a method to check if the character is a vowel or consonant and return the result. The logic used here is as follows:
      i. Convert the character to lowercase if it is an uppercase letter using the ASCII values of the characters
      ii. Check if the character is a vowel or consonant and return Vowel, Consonant, or Not a Letter
   b. Create a Method to Method to find vowels and consonants in a string using charAt() method and finally return the count of vowels and consonants in an array

c. Finally, the main function takes user inputs, calls the user-defined methods, and displays the result.

```
package Day5.LabPractice_L2;
import java.util.Scanner;
public class LP5 {
   public static String checkCharacterType(char ch) {
       ch = (ch >= 'A' && ch <= 'Z') ? (char) (ch + 32) : ch;
       if (ch >= 'a' && ch <= 'z') {
           if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch ==
'u') {
               return "Vowel";
           } else {
               return "Consonant";
           }
       }
       return "Not a Letter";
   }
   public static int[] countVowelsAndConsonants(String text) {
       int vowels = 0, consonants = 0;
       for (int i = 0; i < text.length(); i++) {
           String type = checkCharacterType(text.charAt(i));
           if (type.equals("Vowel")) {
               vowels++;
           } else if (type.equals("Consonant")) {
               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 inputText = scanner.nextLine();
```

```
        int[] counts = countVowelsAndConsonants(inputText);
        System.out.println("\nVowel Count: " + counts[0]);
        System.out.println("Consonant Count: " + counts[1]);
        scanner.close();
    }
}
```

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

   Hint =>

   a. Create a method to check if the character is a vowel or consonant and return the result. The logic used here is as follows:
      i. Convert the character to lowercase if it is an uppercase letter using the ASCII values of the characters
      ii. Check if the character is a vowel or consonant and return Vowel, Consonant, or Not a Letter
   b. Create a Method to find vowels and consonants in a string using charAt() method and return the character and vowel or consonant in a 2D array
   c. Create a Method to display the 2D Array of Strings in a Tabular Format
   d. Finally, the main function takes user inputs, calls the user-defined methods, and displays the result.

```
package Day5.LabPractice_L2;
import java.util.Scanner;
public class LP6 {
    public static String classifyCharacter(char ch) {
        ch = (ch >= 'A' && ch <= 'Z') ? (char) (ch + 32) : ch;
        if (ch >= 'a' && ch <= 'z') {
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 
'u') {
                return "Vowel";
            } else {
                return "Consonant";
            }
        }
        return "Not a Letter";
```

```
    }
    public static String[][] analyzeString(String text) {
        int length = text.length();
        String[][] result = new String[length][2];
        for (int i = 0; i < length; i++) {
            char ch = text.charAt(i);
            result[i][0] = String.valueOf(ch);
            result[i][1] = classifyCharacter(ch);
        }
        return result;
    }
    public static void displayResult(String[][] data) {
        System.out.println("\nCharacter\tType");
        System.out.println("----------------------------");
        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 inputText = scanner.nextLine();
        String[][] classifiedData = analyzeString(inputText);
        displayResult(classifiedData);
        scanner.close();
    }
}
```

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

   Hint =>

   a. Create a method to trim the leading and trailing spaces from a string using the ***charAt()*** method. Inside the method run a couple of loops to trim leading and trailing spaces and

determine the starting and ending points with no spaces. Return the start point and end point in an array

b. Write a method to create a substring from a string using the charAt() method with the string, start, and end index as the parameters
c. Write a method to compare two strings using the charAt() method and return a boolean result
d. The main function calls the user-defined trim and substring methods to get the text after trimming the leading and trailing spaces. Post that use the String built-in method ***trim()*** to trim spaces and compare the two strings. And finally display the result

```
package Day5.LabPractice_L2;

import java.util.Scanner;

public class LP7 {

   public static int[] findTrimIndexes(String text) {

       int start = 0, end = text.length() - 1;

       while (start <= end && text.charAt(start) == ' ') {

           start++;

       }

       while (end >= start && text.charAt(end) == ' ') {

           end--;

       }

       return new int[]{start, end + 1};

   }

   public static String customSubstring(String text, int start, int end)
{

       StringBuilder subStr = new StringBuilder();

       for (int i = start; i < end; i++) {

           subStr.append(text.charAt(i));

       }

       return subStr.toString();

   }

   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 spaces: ");
        String inputText = scanner.nextLine();
        int[] indexes = findTrimIndexes(inputText);
        String customTrimmed = customSubstring(inputText, indexes[0],
indexes[1]);
        String builtInTrimmed = inputText.trim();
        boolean areEqual = compareStrings(customTrimmed, builtInTrimmed);
        System.out.println("\nOriginal String: '" + inputText + "'");
        System.out.println("Custom Trimmed: '" + customTrimmed + "'");
        System.out.println("Built-in Trimmed: '" + builtInTrimmed + "'");
        System.out.println("Are both methods equal? " + areEqual);
        scanner.close();
    }
}
```

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.

   Hint =>

   a. Create a method to define the random 2-digit age of several students provided as method parameters and return a 1D array of ages of n students
   b. Create a method that takes an array of age as a parameter and returns a 2D String array of age and a boolean true or false to indicate can and cannot vote. Inside the method firstly validate the age for a negative number, if a negative cannot vote. For valid age check for age is 18 or above to set true to indicate can vote.
   c. Create a method to display the 2D array in a tabular format.
   d. Finally, the main function takes user inputs, calls the user-defined methods, and displays the result.

```java
package Day5.LabPractice_L2;
import java.util.Random;
import java.util.Scanner;
public class LP8 {
    public static int[] generateAges(int numStudents) {
        Random random = new Random();
        int[] ages = new int[numStudents];
        for (int i = 0; i < numStudents; i++) {
            ages[i] = random.nextInt(50) - 10;
        }
        return ages;
    }
    public static String[][] checkVotingEligibility(int[] ages) {
        String[][] results = new String[ages.length][2];
        for (int i = 0; i < ages.length; i++) {
            results[i][0] = String.valueOf(ages[i]);
            if (ages[i] < 0) {
                results[i][1] = "Invalid Age";
            } else if (ages[i] >= 18) {
                results[i][1] = "Can Vote";
            } else {
                results[i][1] = "Cannot Vote";
            }
        }
        return results;
    }
    public static void displayResults(String[][] data) {
        System.out.println("\nStudent Age Voting Eligibility");
        System.out.println("-------------------------------");
        System.out.printf("%-10s %-15s\n", "Age", "Voting Status");
        System.out.println("-------------------------------");
        for (String[] row : data) {
            System.out.printf("%-10s %-15s\n", row[0], row[1]);
```

```
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int numStudents = scanner.nextInt();
        int[] ages = generateAges(numStudents);
        String[][] results = checkVotingEligibility(ages);
        displayResults(results);
        scanner.close();
    }
}
```

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.

   Hint =>

   a. **The rule is:** rock-scissors: rock will win (rock crushes scissors); rock-paper: paper wins (paper covers rock); scissors-paper: scissors win (scissors cuts paper)
   b. Create a Method to find the Computer Choice using the Math.random
   c. Create a Method to find the winner between the user and the computer
   d. Create a Method to find the average and percentage of wins for the user and the computer and return a String 2D array
   e. Create a Method to display the results of every game and also display the average and percentage wins
   f. In the main take user input for the number of games and call methods to display results

```
package Day5.LabPractice_L2;
import java.util.Random;
import java.util.Scanner;
public class LP9 {
    public static String getComputerChoice() {
        String[] choices = {"Rock", "Paper", "Scissors"};
        int index = (int) (Math.random() * 3);
        return choices[index];
```

```
    }

    public static String determineWinner(String playerChoice, String
computerChoice) {
        if (playerChoice.equalsIgnoreCase(computerChoice)) {
            return "Draw";
        } else if (
                (playerChoice.equalsIgnoreCase("Rock") &&
computerChoice.equalsIgnoreCase("Scissors")) ||
                        (playerChoice.equalsIgnoreCase("Paper") &&
computerChoice.equalsIgnoreCase("Rock")) ||
                        (playerChoice.equalsIgnoreCase("Scissors") &&
computerChoice.equalsIgnoreCase("Paper"))
        ) {
            return "Player Wins";
        } else {
            return "Computer Wins";
        }
    }
    public static String[][] calculateStats(int playerWins, int
computerWins, int totalGames) {
        String[][] stats = new String[2][2];
        stats[0][0] = "Player Wins";
        stats[0][1] = String.format("%.2f%%", ((double) playerWins /
totalGames) * 100);
        stats[1][0] = "Computer Wins";
        stats[1][1] = String.format("%.2f%%", ((double) computerWins /
totalGames) * 100);
        return stats;
    }
    public static void displayResults(String[][] results, String[][]
stats) {
        System.out.println("\nGame Results:");
        System.out.println("---------------------------------");
        System.out.printf("%-10s %-12s %-15s\n", "Game", "Player",
"Computer");
        System.out.println("---------------------------------");
```

```
        for (int i = 0; i < results.length; i++) {
            System.out.printf("%-10s %-12s %-15s\n", results[i][0],
results[i][1], results[i][2]);
        }
        System.out.println("\nWin Statistics:");
        System.out.println("------------------------------");
        System.out.printf("%-15s %-10s\n", "Category", "Win %");
        System.out.println("------------------------------");
        for (String[] row : stats) {
            System.out.printf("%-15s %-10s\n", row[0], row[1]);
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of games to play: ");
        int numGames = scanner.nextInt();
        scanner.nextLine();
        String[][] results = new String[numGames][3];
        int playerWins = 0, computerWins = 0;
        for (int i = 0; i < numGames; i++) {
            System.out.print("\nEnter Rock, Paper, or Scissors: ");
            String playerChoice = scanner.nextLine();
            while (!playerChoice.equalsIgnoreCase("Rock") &&
                    !playerChoice.equalsIgnoreCase("Paper") &&
                    !playerChoice.equalsIgnoreCase("Scissors")) {
                System.out.print("Invalid choice. Please enter Rock,
Paper, or Scissors: ");
                playerChoice = scanner.nextLine();
            }
            String computerChoice = getComputerChoice();
            String result = determineWinner(playerChoice, computerChoice);
            if (result.equals("Player Wins")) {
                playerWins++;
```

```
            } else if (result.equals("Computer Wins")) {
                computerWins++;
            }
            results[i][0] = "Game " + (i + 1);
            results[i][1] = playerChoice;
            results[i][2] = computerChoice;
        }
        String[][] stats = calculateStats(playerWins, computerWins,
numGames);
        displayResults(results, stats);
        scanner.close();
    }
}
```

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

Grade	Remarks	Marks
A	(Level 4, above agency-normalized standards)	80% and above
B	(Level 3, at agency-normalized standards)	70-79%
C	(Level 2, below, but approaching agency-normalized standards)	60-69%
D	(Level 1, well below agency-normalized standards)	50-59%
E	(Level 1- , too below agency-normalized standards)	40-49%
R	(Remedial standards)	39% and below

Hint =>

a. Write a method to generate random 2-digit scores for Physics, Chemistry and Math (PCM) for the students and return the scores. This method returns a 2D array with PCM scores for all students
b. Write a Method to calculate the total, average, and percentages for each student and return a 2D array with the corresponding values. Please ensure to round off the values to 2 Digits using ***Math.round()*** method
c. Write a Method to calculate the grade based on the percentage as shown in the ref table and return a 2D array of students' grade

d. Finally write a Method to display the scorecard of all students with their scores, total, average, percentage, and grade in a tabular format.

```
package Day5.LabPractice_L2;
import java.util.Random;
import java.util.Scanner;
public class LP10 {
   public static int[][] generateMarks(int numStudents) {
       Random rand = new Random();
       int[][] marks = new int[numStudents][3];
       for (int i = 0; i < numStudents; i++) {
           marks[i][0] = rand.nextInt(41) + 60;
           marks[i][1] = rand.nextInt(41) + 60;
           marks[i][2] = rand.nextInt(41) + 60;
       }
       return marks;
   }
   public static double[][] calculateScores(int[][] marks) {
       int numStudents = marks.length;
       double[][] results = new double[numStudents][3];
       for (int i = 0; i < numStudents; i++) {
           int total = marks[i][0] + marks[i][1] + marks[i][2];
           double average = total / 3.0;
           double percentage = Math.round((total / 3.0) * 100.0) / 100.0;
           results[i][0] = total;
           results[i][1] = average;
           results[i][2] = percentage;
       }
       return results;
   }
   public static String[][] assignGrades(double[][] scores) {
       int numStudents = scores.length;
       String[][] grades = new String[numStudents][2];
       for (int i = 0; i < numStudents; i++) {
```

```
            double percentage = scores[i][2];
            if (percentage >= 80) {
                grades[i][0] = "A";
                grades[i][1] = "Level-4, above agency-normalised
standards";
            } else if (percentage >= 70) {
                grades[i][0] = "B";
                grades[i][1] = "Level-3, at agency-normalised standards";
            } else if (percentage >= 60) {
                grades[i][0] = "C";
                grades[i][1] = "Level-2, below but approaching
agency-normalised standards";
            } else if (percentage >= 50) {
                grades[i][0] = "D";
                grades[i][1] = "Level-1, well below agency-normalised
standards";
            } else if (percentage >= 40) {
                grades[i][0] = "E";
                grades[i][1] = "Level-1, too below agency-normalised
standards";
            } else {
                grades[i][0] = "R";
                grades[i][1] = "Remedial Standards";
            }
        }
        return grades;
    }
    public static void displayResults(int[][] marks, double[][] scores,
String[][] grades) {
        System.out.println("\nStudent Scorecard:");

System.out.println("-------------------------------------------------------
----------------------");
        System.out.printf("%-10s %-10s %-10s %-10s %-10s %-10s %-10s
%-10s\n",
```

```
                "Student", "Physics", "Chemistry", "Math", "Total",
"Average", "Percentage", "Grade");

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


        for (int i = 0; i < marks.length; i++) {
            System.out.printf("%-10d %-10d %-10d %-10d %-10.0f %-10.2f
%-10.2f %-10s\n",
                    (i + 1), marks[i][0], marks[i][1], marks[i][2],
scores[i][0], scores[i][1], scores[i][2], grades[i][0]);
        }
        System.out.println("\nRemarks:");
        for (int i = 0; i < marks.length; i++) {
            System.out.printf("Student %d: %s\n", (i + 1), grades[i][1]);
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int numStudents = scanner.nextInt();
        int[][] marks = generateMarks(numStudents);
        double[][] scores = calculateScores(marks);
        String[][] grades = assignGrades(scores);
        displayResults(marks, scores, grades);
        scanner.close();
    }
}
```