

## Level 2 String

```
import java.util.*;
```

### 1. Find Length of String

```
class StringLength {  
    public static int findLength(String text) {  
        int count = 0;  
        try {  
            while (true) {  
                text.charAt(count);  
                count++;  
            }  
        } catch (StringIndexOutOfBoundsException e) {}  
        return count;  
    }  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter a string: ");  
        String input = sc.next();  
  
        System.out.println("Custom Length: " + findLength(input));  
        System.out.println("Built-in Length: " + input.length());  
    }  
}
```

### 2. Split Text into Words and Compare

```
class CustomSplit {  
    public static String[] splitWords(String text) {
```

```

ArrayList<String> words = new ArrayList<>();
StringBuilder word = new StringBuilder();
for (int i = 0; i < text.length(); i++) {
    char c = text.charAt(i);
    if (c == ' ') {
        if (word.length() > 0) {
            words.add(word.toString());
            word.setLength(0);
        }
    } else {
        word.append(c);
    }
}
if (word.length() > 0) words.add(word.toString());
return words.toArray(new String[0]);
}

```

```

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 sc = new Scanner(System.in);
    System.out.print("Enter a sentence: ");
    String input = sc.nextLine();
}

```

```

String[] customSplit = splitWords(input);
String[] builtInSplit = input.split(" ");

System.out.println("Custom Split: " + Arrays.toString(customSplit));
System.out.println("Built-in Split: " + Arrays.toString(builtInSplit));
System.out.println("Arrays Equal? " + compareArrays(customSplit, builtInSplit));
}
}

```

### 3. Words with Their Lengths

```

class WordsWithLength {
    public static String[] splitWords(String text) {
        ArrayList<String> words = new ArrayList<>();
        StringBuilder word = new StringBuilder();
        for (int i = 0; i < text.length(); i++) {
            char c = text.charAt(i);
            if (c == ' ') {
                if (word.length() > 0) {
                    words.add(word.toString());
                    word.setLength(0);
                }
            } else {
                word.append(c);
            }
        }
        if (word.length() > 0) words.add(word.toString());
        return words.toArray(new String[0]);
    }
}

```

```

public static String[][] wordWithLength(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(words[i].length());
    }
    return result;
}

```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a sentence: ");
    String input = sc.nextLine();

    String[] words = splitWords(input);
    String[][] result = wordWithLength(words);

    System.out.println("Word\tLength");
    for (String[] row : result) {
        System.out.println(row[0] + "\t" + row[1]);
    }
}

```

#### 4. Shortest and Longest Word

```

class ShortestLongest {
    public static String[] splitWords(String text) {
        ArrayList<String> words = new ArrayList<>();
    }
}

```

```

StringBuilder word = new StringBuilder();
for (int i = 0; i < text.length(); i++) {
    char c = text.charAt(i);
    if (c == ' ') {
        if (word.length() > 0) {
            words.add(word.toString());
            word.setLength(0);
        }
    } else {
        word.append(c);
    }
}
if (word.length() > 0) words.add(word.toString());
return words.toArray(new String[0]);
}

```

```

public static int[] findShortestLongest(String[] words) {
    int minIndex = 0, maxIndex = 0;
    for (int i = 1; i < words.length; i++) {
        if (words[i].length() < words[minIndex].length()) minIndex = i;
        if (words[i].length() > words[maxIndex].length()) maxIndex = i;
    }
    return new int[]{minIndex, maxIndex};
}

```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a sentence: ");
    String input = sc.nextLine();
}

```

```

String[] words = splitWords(input);
int[] result = findShortestLongest(words);

System.out.println("Shortest Word: " + words[result[0]]);
System.out.println("Longest Word: " + words[result[1]]);
}
}

```

## 5. Count Vowels and Consonants

```

class VowelConsonantCount {

    public static boolean isVowel(char c) {

        c = Character.toLowerCase(c);

        return "aeiou".indexOf(c) != -1;

    }

    public static int[] countVowelsConsonants(String text) {

        int vowels = 0, consonants = 0;

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

            char c = text.charAt(i);

            if (Character.isLetter(c)) {

                if (isVowel(c)) vowels++;

                else consonants++;

            }

        }

        return new int[]{vowels, consonants};

    }

    public static void main(String[] args) {

```

```

Scanner sc = new Scanner(System.in);
System.out.print("Enter a string: ");
String input = sc.nextLine();

int[] result = countVowelsConsonants(input);
System.out.println("Vowels: " + result[0] + ", Consonants: " + result[1]);
}
}

```

## 6. Display Character Type

```

class CharacterType {
    public static String charType(char c) {
        c = Character.toLowerCase(c);
        if ("aeiou".indexOf(c) != -1) return "Vowel";
        else if (c >= 'a' && c <= 'z') return "Consonant";
        else return "Not a Letter";
    }

    public static String[][] analyzeString(String text) {
        String[][] result = new String[text.length()][2];
        for (int i = 0; i < text.length(); i++) {
            result[i][0] = String.valueOf(text.charAt(i));
            result[i][1] = charType(text.charAt(i));
        }
        return result;
    }

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

```

```

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

String[][] result = analyzeString(input);
System.out.println("Char\tType");
for (String[] row : result) {
    System.out.println(row[0] + "\t" + row[1]);
}
}
}

```

## 7. Trim Leading and Trailing Spaces

```

class CustomTrim {

    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};
    }

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

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string with spaces: ");
    }
}

```



```

String input = sc.nextLine();

int[] indexes = findTrimIndexes(input);

String trimmed = substring(input, indexes[0], indexes[1]);

System.out.println("Custom Trim: " + trimmed + "");
System.out.println("Built-in Trim: " + input.trim() + "");
}
}

```

## 8. Voting Eligibility

```

class Voting {

    public static int[] generateAges(int n) {

        Random rand = new Random();

        int[] ages = new int[n];

        for (int i = 0; i < n; i++) ages[i] = rand.nextInt(90) + 10;

        return ages;

    }

    public static String[][] canVote(int[] ages) {

        String[][] result = new String[ages.length][2];

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

            result[i][0] = String.valueOf(ages[i]);

            result[i][1] = (ages[i] >= 18) ? "true" : "false";

        }

        return result;

    }

    public static void main(String[] args) {

```

```

int[] ages = generateAges(10);
String[][] result = canVote(ages);

System.out.println("Age\tCan Vote");
for (String[] row : result) {
    System.out.println(row[0] + "\t" + row[1]);
}
}
}

```

## 9. Rock Paper Scissors Game

```

class RockPaperScissors {

    public static String computerChoice() {
        String[] choices = {"rock", "paper", "scissors"};
        return choices[(int)(Math.random() * 3)];
    }

    public static int findWinner(String user, String comp) {
        if (user.equals(comp)) return 0;
        if ((user.equals("rock") && comp.equals("scissors")) ||
            (user.equals("paper") && comp.equals("rock")) ||
            (user.equals("scissors") && comp.equals("paper"))) return 1;
        return -1;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of games: ");
        int games = sc.nextInt();
    }
}

```

```

sc.nextLine();

int userWins = 0, compWins = 0;
for (int i = 0; i < games; i++) {
    System.out.print("Enter rock/paper/scissors: ");
    String user = sc.nextLine().toLowerCase();
    String comp = computerChoice();
    int winner = findWinner(user, comp);
    System.out.println("Computer chose: " + comp);
    if (winner == 1) { System.out.println("You win!"); userWins++; }
    else if (winner == -1) { System.out.println("Computer wins!"); compWins++; }
    else System.out.println("Draw!");
}

System.out.println("Final Stats:");
System.out.println("User Wins: " + userWins + " (" + (userWins * 100 / games) + "%)");
System.out.println("Computer Wins: " + compWins + " (" + (compWins * 100 / games) +
"%)");
}
}

```

## 10. Student Marks and Grades

```

class StudentMarks {
    public static int[][] generateMarks(int n) {
        Random rand = new Random();
        int[][] marks = new int[n][3];
        for (int i = 0; i < n; i++) {
            marks[i][0] = rand.nextInt(41) + 60; // Physics
            marks[i][1] = rand.nextInt(41) + 60; // Chemistry
            marks[i][2] = rand.nextInt(41) + 60; // Maths

```

```
}  
    return marks;  
}
```

```
public static String grade(double percent) {  
    if (percent >= 90) return "A";  
    else if (percent >= 75) return "B";  
    else if (percent >= 60) return "C";  
    else if (percent >= 40) return "D";  
    else return "F";  
}
```

```
public static void main(String[] args) {  
    int[][] marks = generateMarks(5);  
    System.out.println("Phy\tChem\tMath\tTotal\tAvg\tPercent\tGrade");  
  
    for (int[] m : marks) {  
        int total = m[0] + m[1] + m[2];  
        double avg = total / 3.0;  
        double percent = (total / 300.0) * 100;  
        System.out.println(m[0] + "\t" + m[1] + "\t" + m[2] + "\t" +  
            total + "\t" + String.format("%.2f", avg) + "\t" +  
            String.format("%.2f", percent) + "\t" + grade(percent));  
    }  
}
```