Week 1 - S1 - Practice Problem Solution

Ramesh Harisabapathi Chettiar

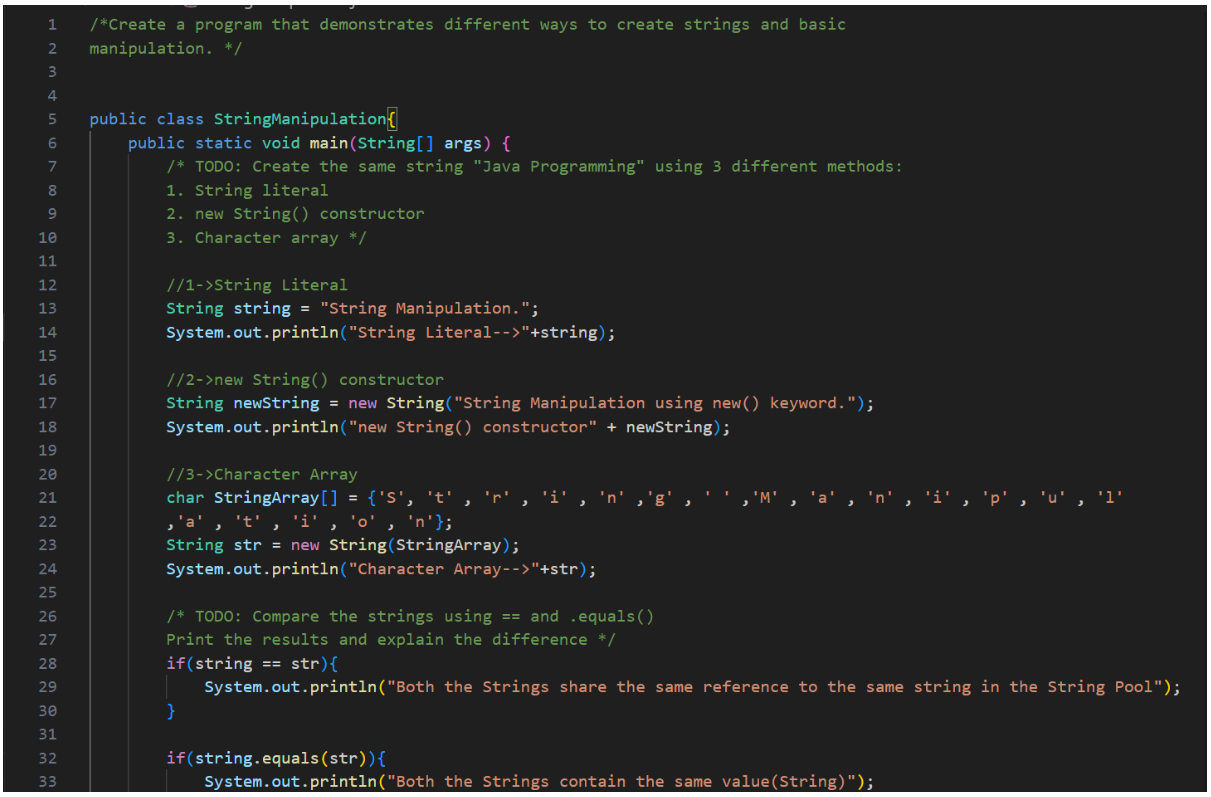
RA2411030010263

Date of Submission:17/08/2025

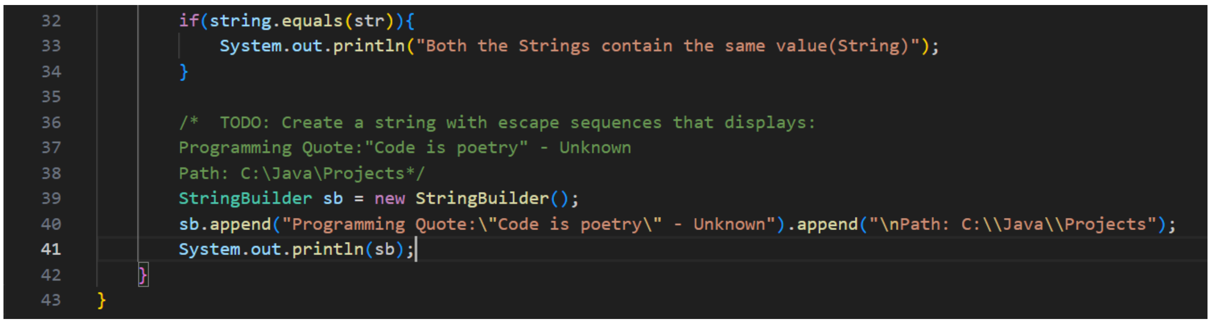
🛠 PRACTICE PROBLEM 1:

String Creation and Manipulation

Task: Create a program that demonstrates different ways to create strings and basic manipulation.



```java
/*Create a program that demonstrates different ways to create strings and basic
manipulation. */


public class StringManipulation{
    public static void main(String[] args) {
        /* TODO: Create the same string "Java Programming" using 3 different methods:
        1. String literal
        2. new String() constructor
        3. Character array */

        //1->String Literal
        String string = "String Manipulation.";
        System.out.println("String Literal-->"+string);

        //2->new String() constructor
        String newString = new String("String Manipulation using new() keyword.");
        System.out.println("new String() constructor" + newString);

        //3->Character Array
        char StringArray[] = {'S', 't' , 'r' , 'i' , 'n' ,'g' , ' ' ,'M' , 'a' , 'n' , 'i' , 'p' , 'u' , 'l'
        ,'a' , 't' , 'i' , 'o' , 'n'};
        String str = new String(StringArray);
        System.out.println("Character Array-->"+str);

        /* TODO: Compare the strings using == and .equals()
        Print the results and explain the difference */
        if(string == str){
            System.out.println("Both the Strings share the same reference to the same string in the String Pool");
        }

        if(string.equals(str)){
            System.out.println("Both the Strings contain the same value(String)");
```

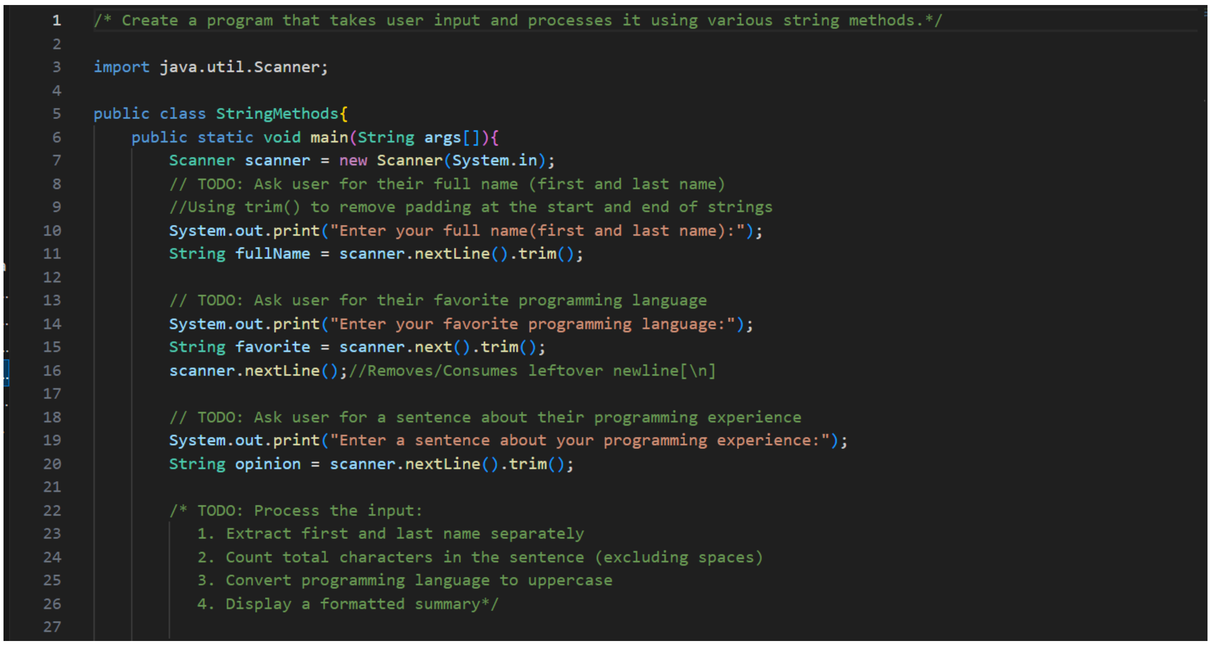


```java
        if(string.equals(str)){
            System.out.println("Both the Strings contain the same value(String)");
        }

        /*  TODO: Create a string with escape sequences that displays:
        Programming Quote:"Code is poetry" - Unknown
        Path: C:\Java\Projects*/
        StringBuilder sb = new StringBuilder();
        sb.append("Programming Quote:\"Code is poetry\" - Unknown").append("\nPath: C:\\Java\\Projects");
        System.out.println(sb);
    }
}
```

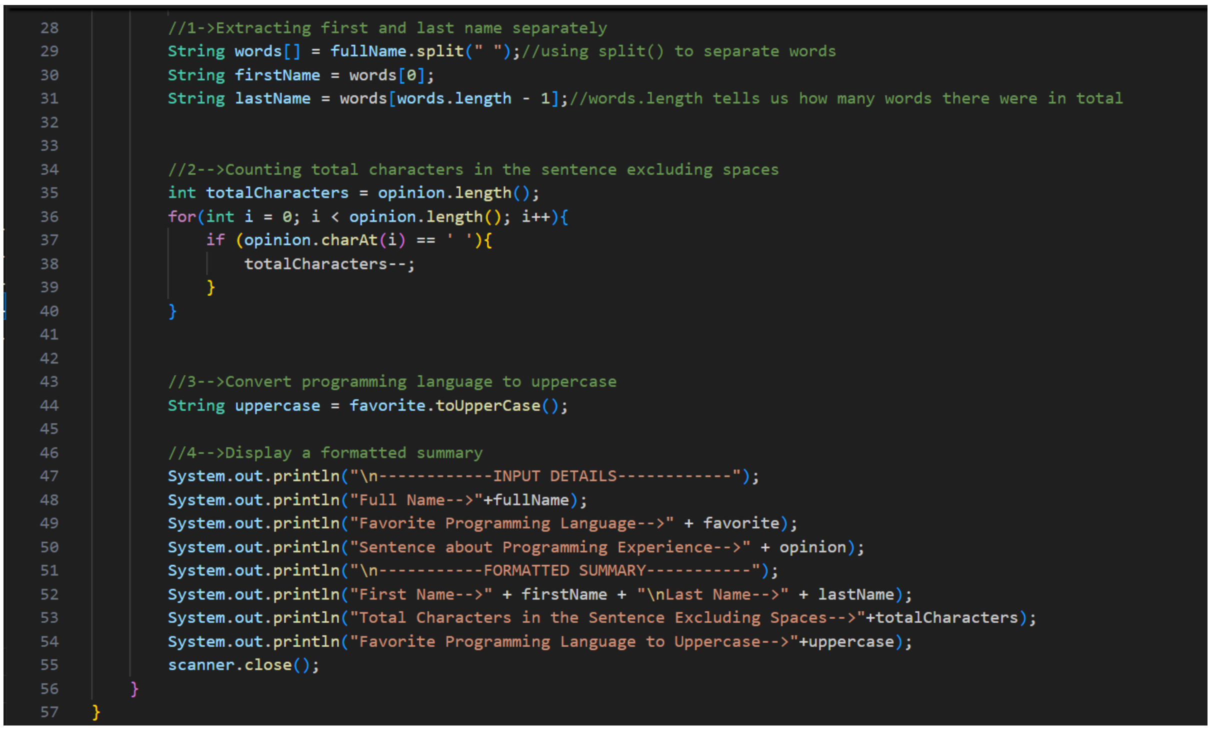
🛠 PRACTICE PROBLEM 2:

String Input and Processing

Task: Create a program that takes user input and processes it using various string methods.



```java
/* Create a program that takes user input and processes it using various string methods.*/

import java.util.Scanner;

public class StringMethods{
    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);
        // TODO: Ask user for their full name (first and last name)
        //Using trim() to remove padding at the start and end of strings
        System.out.print("Enter your full name(first and last name):");
        String fullName = scanner.nextLine().trim();

        // TODO: Ask user for their favorite programming language
        System.out.print("Enter your favorite programming language:");
        String favorite = scanner.next().trim();
        scanner.nextLine();//Removes/Consumes leftover newline[\n]

        // TODO: Ask user for a sentence about their programming experience
        System.out.print("Enter a sentence about your programming experience:");
        String opinion = scanner.nextLine().trim();

        /* TODO: Process the input:
            1. Extract first and last name separately
            2. Count total characters in the sentence (excluding spaces)
            3. Convert programming language to uppercase
            4. Display a formatted summary*/

```

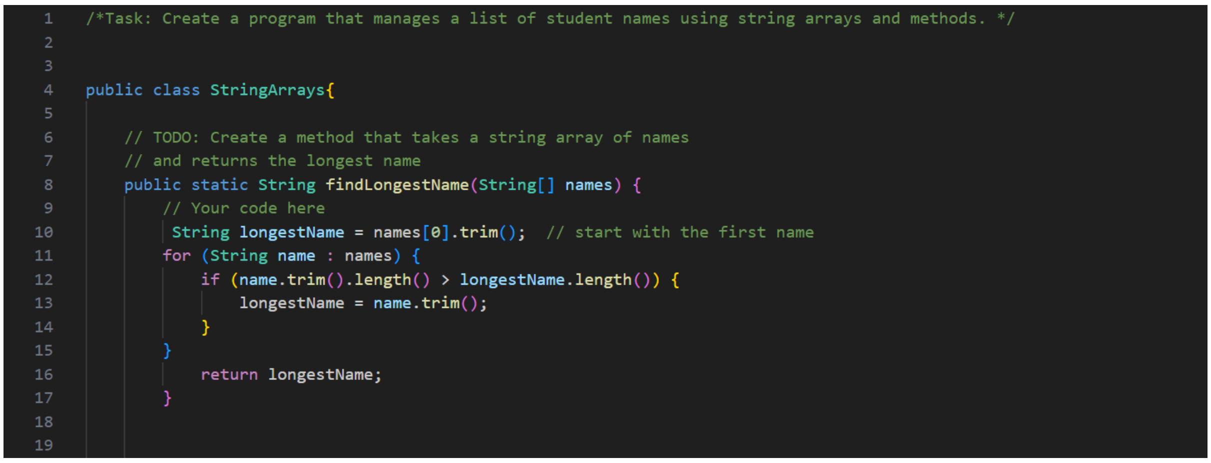


```java
        //1->Extracting first and last name separately
        String words[] = fullName.split(" ");//using split() to separate words
        String firstName = words[0];
        String lastName = words[words.length - 1];//words.length tells us how many words there were in total


        //2-->Counting total characters in the sentence excluding spaces
        int totalCharacters = opinion.length();
        for(int i = 0; i < opinion.length(); i++){
            if (opinion.charAt(i) == ' '){
                totalCharacters--;
            }
        }


        //3-->Convert programming language to uppercase
        String uppercase = favorite.toUpperCase();

        //4-->Display a formatted summary
        System.out.println("\n-----------INPUT DETAILS-----------");
        System.out.println("Full Name-->"+fullName);
        System.out.println("Favorite Programming Language-->" + favorite);
        System.out.println("Sentence about Programming Experience-->" + opinion);
        System.out.println("\n----------FORMATTED SUMMARY----------");
        System.out.println("First Name-->" + firstName + "\nLast Name-->" + lastName);
        System.out.println("Total Characters in the Sentence Excluding Spaces-->"+totalCharacters);
        System.out.println("Favorite Programming Language to Uppercase-->"+uppercase);
        scanner.close();
    }
}
```

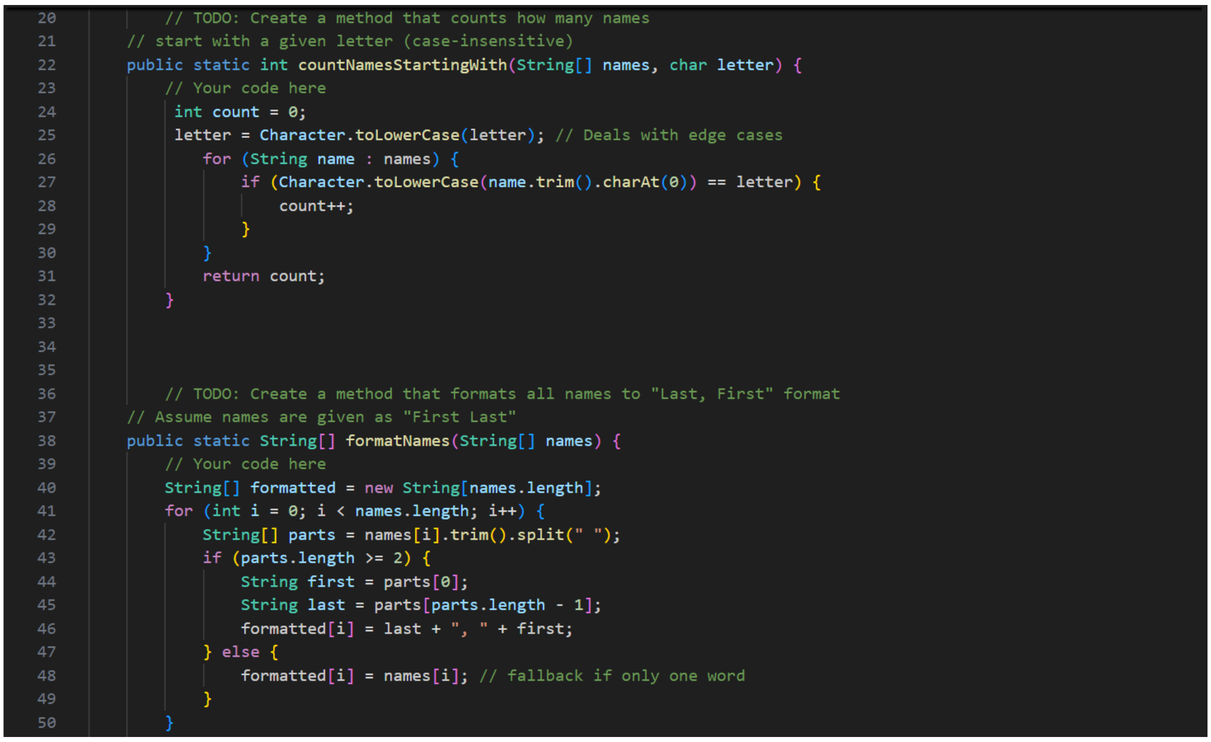
🛠 PRACTICE PROBLEM 3:

String Arrays and Methods

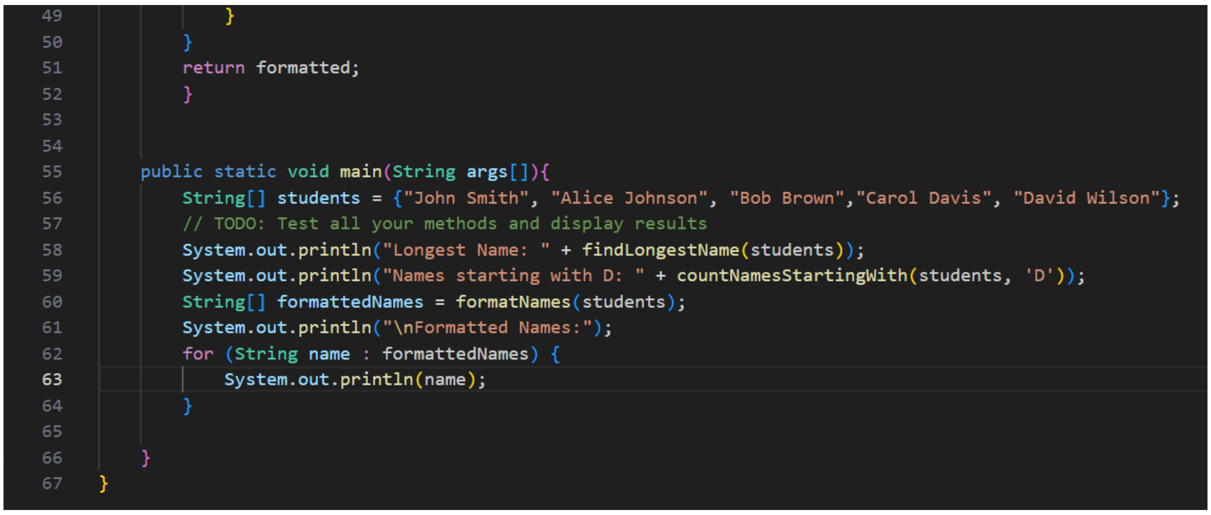
Task: Create a program that manages a list of student names using string arrays and methods.



```java
/*Task: Create a program that manages a list of student names using string arrays and methods. */


public class StringArrays{

    // TODO: Create a method that takes a string array of names
    // and returns the longest name
    public static String findLongestName(String[] names) {
        // Your code here
        String longestName = names[0].trim();  // start with the first name
        for (String name : names) {
            if (name.trim().length() > longestName.length()) {
                longestName = name.trim();
            }
        }
            return longestName;
        }


```



```java
        // TODO: Create a method that counts how many names
    // start with a given letter (case-insensitive)
    public static int countNamesStartingWith(String[] names, char letter) {
        // Your code here
        int count = 0;
        letter = Character.toLowerCase(letter); // Deals with edge cases
            for (String name : names) {
                if (Character.toLowerCase(name.trim().charAt(0)) == letter) {
                    count++;
                }
            }
            return count;
        }



        // TODO: Create a method that formats all names to "Last, First" format
    // Assume names are given as "First Last"
    public static String[] formatNames(String[] names) {
        // Your code here
        String[] formatted = new String[names.length];
        for (int i = 0; i < names.length; i++) {
            String[] parts = names[i].trim().split(" ");
            if (parts.length >= 2) {
                String first = parts[0];
                String last = parts[parts.length - 1];
                formatted[i] = last + ", " + first;
            } else {
                formatted[i] = names[i]; // fallback if only one word
            }
        }
```

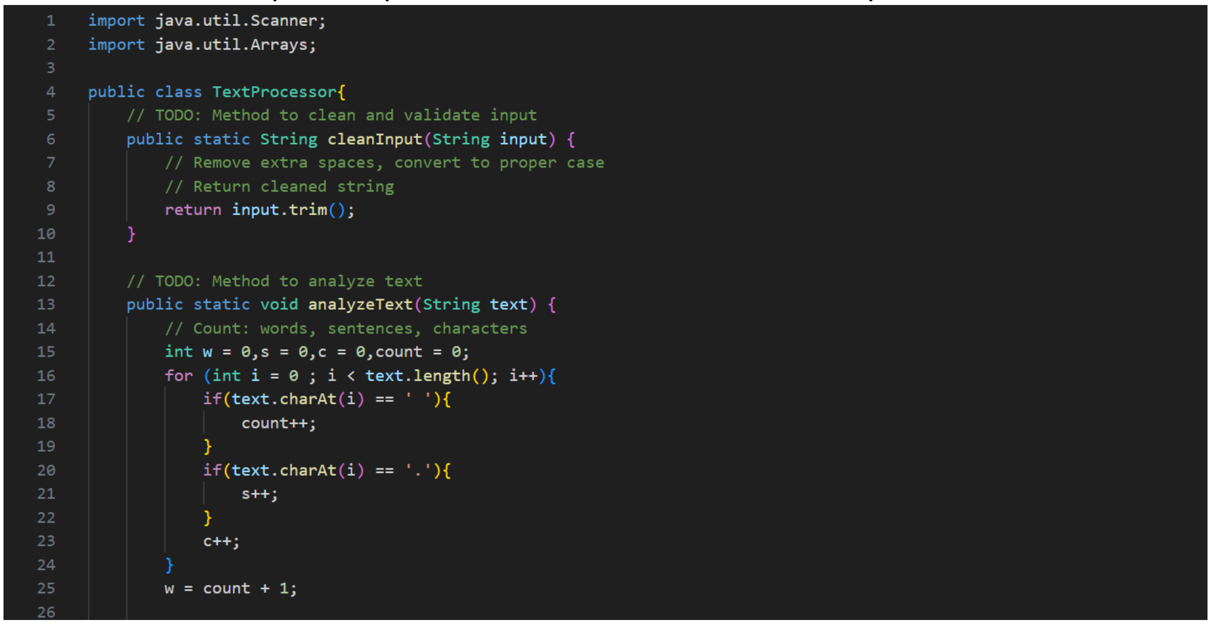


```
            }
        }
        return formatted;
        }


    public static void main(String args[]){
        String[] students = {"John Smith", "Alice Johnson", "Bob Brown","Carol Davis", "David Wilson"};
        // TODO: Test all your methods and display results
        System.out.println("Longest Name: " + findLongestName(students));
        System.out.println("Names starting with D: " + countNamesStartingWith(students, 'D'));
        String[] formattedNames = formatNames(students);
        System.out.println("\nFormatted Names:");
        for (String name : formattedNames) {
            System.out.println(name);
        }

    }
}
```

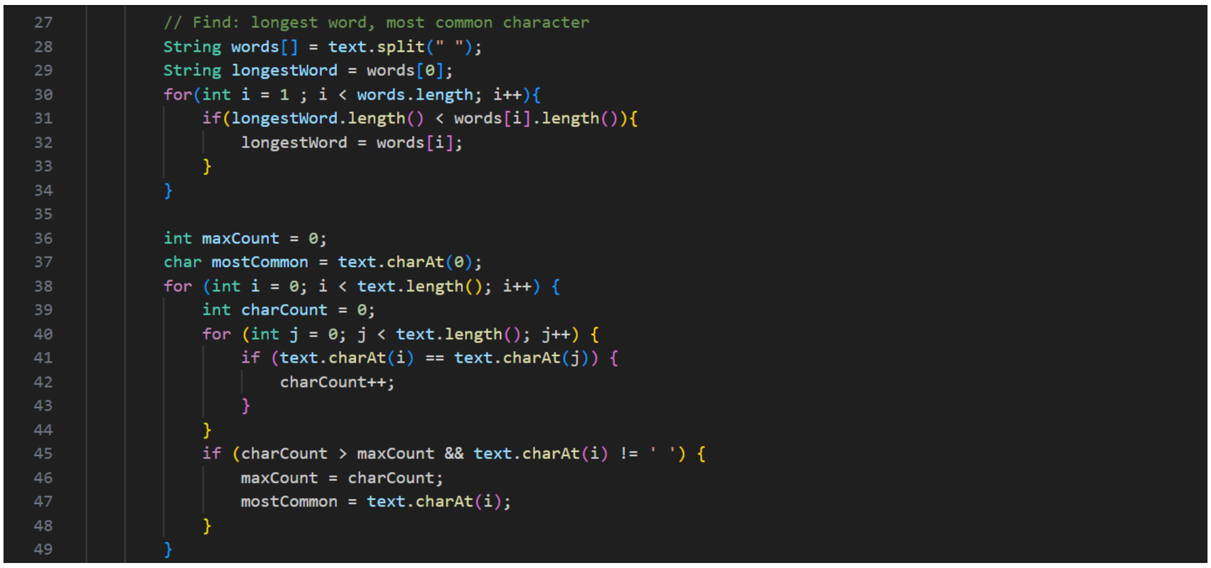
🛠 PRACTICE PROBLEM 4:

Complete String Application (10 minutes)

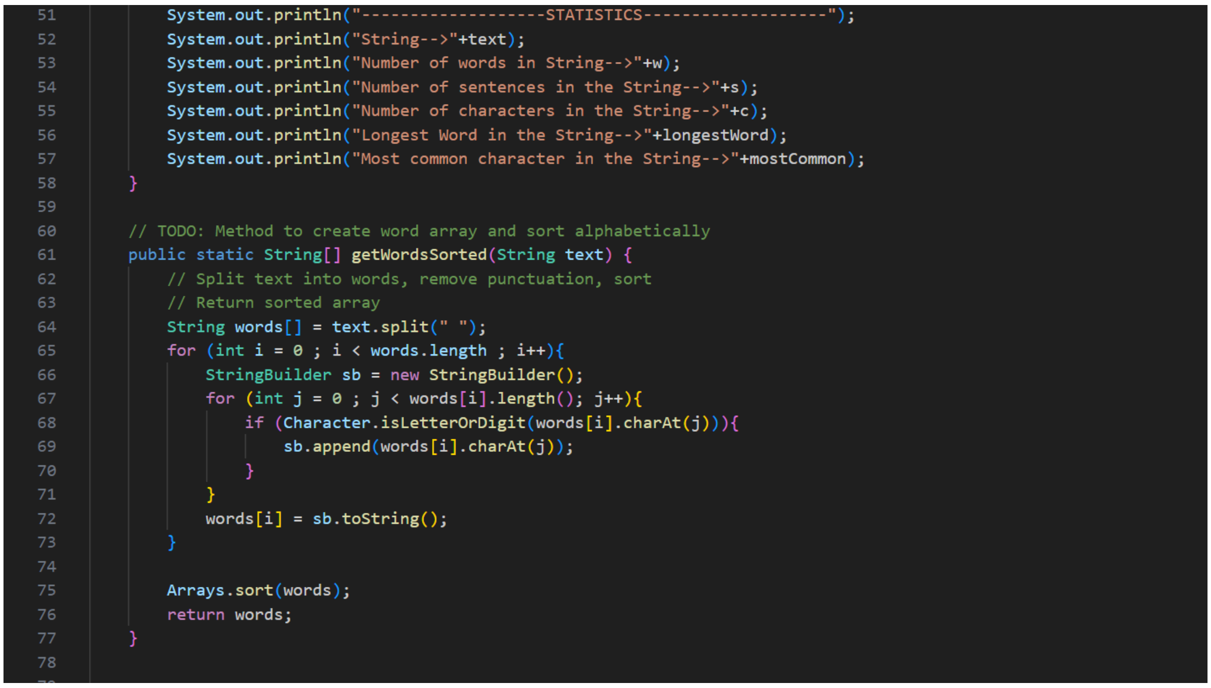
Task: Create a simple text processor that combines all concepts learned.



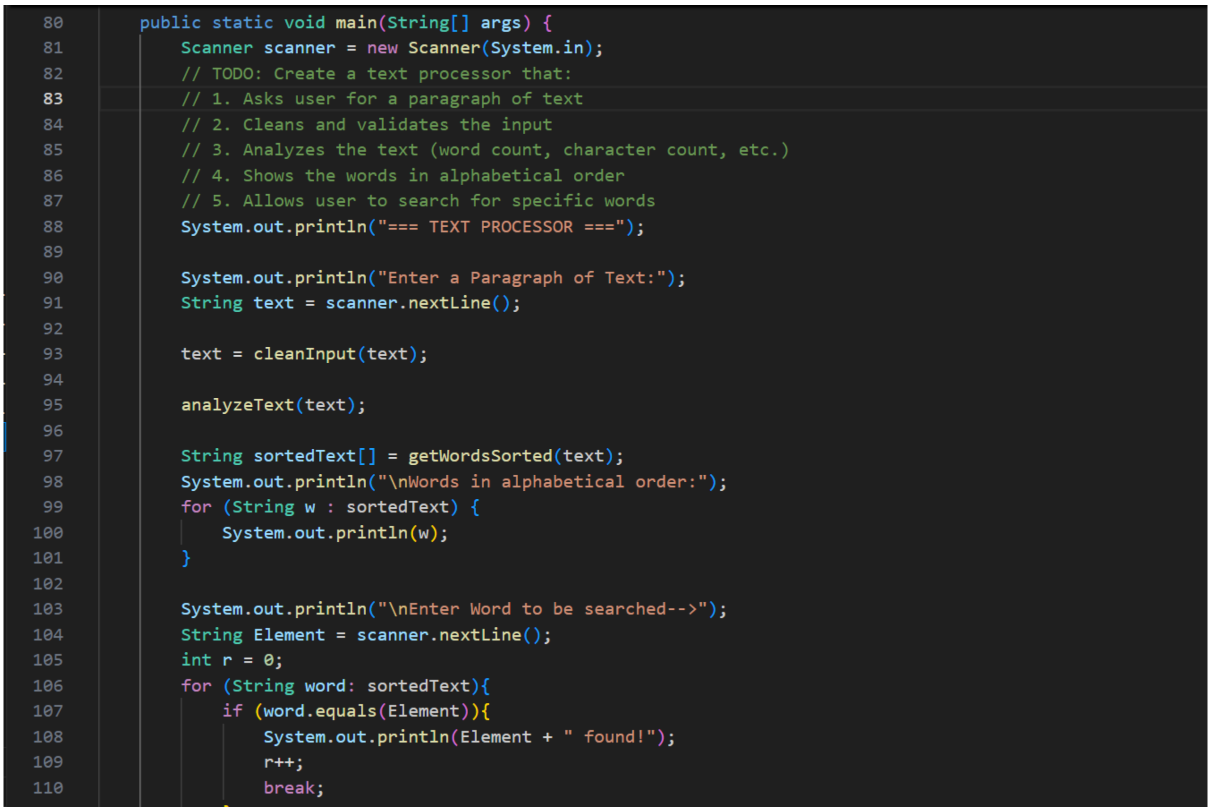
```java
import java.util.Scanner;
import java.util.Arrays;

public class TextProcessor{
    // TODO: Method to clean and validate input
    public static String cleanInput(String input) {
        // Remove extra spaces, convert to proper case
        // Return cleaned string
        return input.trim();
    }

    // TODO: Method to analyze text
    public static void analyzeText(String text) {
        // Count: words, sentences, characters
        int w = 0,s = 0,c = 0,count = 0;
        for (int i = 0 ; i < text.length(); i++){
            if(text.charAt(i) == ' '){
                count++;
            }
            if(text.charAt(i) == '.'){
                s++;
            }
            c++;
        }
        w = count + 1;

```



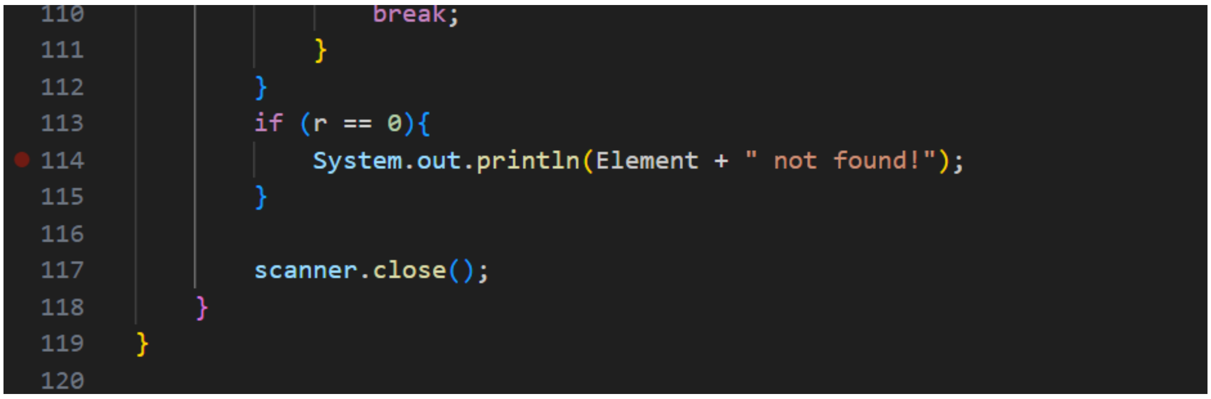
```java
        // Find: longest word, most common character
        String words[] = text.split(" ");
        String longestWord = words[0];
        for(int i = 1 ; i < words.length; i++){
            if(longestWord.length() < words[i].length()){
                longestWord = words[i];
            }
        }

        int maxCount = 0;
        char mostCommon = text.charAt(0);
        for (int i = 0; i < text.length(); i++) {
            int charCount = 0;
            for (int j = 0; j < text.length(); j++) {
                if (text.charAt(i) == text.charAt(j)) {
                    charCount++;
                }
            }
            if (charCount > maxCount && text.charAt(i) != ' ') {
                maxCount = charCount;
                mostCommon = text.charAt(i);
            }
        }
```



```java
        System.out.println("------------------STATISTICS------------------");
        System.out.println("String-->"+text);
        System.out.println("Number of words in String-->"+w);
        System.out.println("Number of sentences in the String-->"+s);
        System.out.println("Number of characters in the String-->"+c);
        System.out.println("Longest Word in the String-->"+longestWord);
        System.out.println("Most common character in the String-->"+mostCommon);
    }

    // TODO: Method to create word array and sort alphabetically
    public static String[] getWordsSorted(String text) {
        // Split text into words, remove punctuation, sort
        // Return sorted array
        String words[] = text.split(" ");
        for (int i = 0 ; i < words.length ; i++){
            StringBuilder sb = new StringBuilder();
            for (int j = 0 ; j < words[i].length(); j++){
                if (Character.isLetterOrDigit(words[i].charAt(j))){
                    sb.append(words[i].charAt(j));
                }
            }
            words[i] = sb.toString();
        }

        Arrays.sort(words);
        return words;
    }

```



```java
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // TODO: Create a text processor that:
        // 1. Asks user for a paragraph of text
        // 2. Cleans and validates the input
        // 3. Analyzes the text (word count, character count, etc.)
        // 4. Shows the words in alphabetical order
        // 5. Allows user to search for specific words
        System.out.println("=== TEXT PROCESSOR ===");

        System.out.println("Enter a Paragraph of Text:");
        String text = scanner.nextLine();

        text = cleanInput(text);

        analyzeText(text);

        String sortedText[] = getWordsSorted(text);
        System.out.println("\nWords in alphabetical order:");
        for (String w : sortedText) {
            System.out.println(w);
        }

        System.out.println("\nEnter Word to be searched-->");
        String Element = scanner.nextLine();
        int r = 0;
        for (String word: sortedText){
            if (word.equals(Element)){
                System.out.println(Element + " found!");
                r++;
                break;
```



```
                    break;
                }
            }
            if (r == 0){
                System.out.println(Element + " not found!");
            }

            scanner.close();
        }
    }

```