

## Assignment-5 for Strings, I/O Operations, and File Management

Subject: CSW2 (CSE 3141)

Session: Jan to May 2025

Branch: CSE

Section: All

Course Outcomes: CO2

Learning Levels: Remembering (L1), Understanding (L2), Application (L3), and Analysis (L4).

Q no.	Questions	Learning Levels
<b>Strings</b>		
Q1.	Write a Java program that illustrates the difference between using string literals and the new keyword for creating <b>String</b> objects. Your program should demonstrate the memory usage implications and how string comparison behaves differently in each case.	L1, L2
Q2.	Write a Java program that demonstrates the immutability of the <b>String</b> class and how it implements the <b>CharSequence</b> interface. Your program should illustrate the behaviours that highlight String immutability and its usage as a CharSequence.	L1, L2
Q3.	Write a Java program that uses <b>StringBuffer</b> to construct a simple text editor which can perform the following operations: a. Append a given string to the existing text. b. Insert a given string at a specified index within the existing text. c. Delete a portion of text between two specified indices. d. Reverse the entire text. e. Replace a portion of the text between two specified indices with a given string. Your program should display a menu with options to perform each of the above operations. After each operation, print the current state of the text. Also, display the current capacity and length of the <b>StringBuffer</b> after each operation to showcase its dynamic nature.	L2, L3
Q4.	Create a Java program that uses <b>StringBuilder</b> to perform a series of text manipulations on a user-provided string. The program should allow users to: a. Add a substring at a specified position. b. Remove a range of characters from the string. c. Modify a character at a specified index. d. Concatenate another string at the end. e. Display the current string after each operation. The program should repeatedly prompt the user to choose an operation until they decide to exit. After each operation, it should display the modified string, demonstrating the mutable nature of <b>StringBuilder</b> .	L2, L3

Q5.	Case Conversion and Comparison: Prompt the user to input two strings. Convert both strings to lowercase and uppercase. Compare the converted strings to check case-insensitive equality. Display the converted strings and the result of the comparison.	L2, L3
Q6.	Character Array and Search: Ask for a string from the user. Convert the string to a character array. Prompt the user to enter a character to search in the string. Find the first and last occurrences of the character. Display the character array and the positions found (if any).	L2, L3
Q7.	Word Replacement in Sentences: Request a sentence and two words from the user: one to search for and one to replace it with. Find the first occurrence of the search word in the sentence. Replace the word using substring operations and concatenation. Display the original and the modified sentences.	L3, L4
Q8.	Interactive String Explorer: Prompt the user for a string. Display a menu with options to perform various operations: convert to lowercase/uppercase, search for a character/index, or concatenate with another string. Based on user selection, perform the appropriate string operation and show the result.	L2, L3
<b>I/O Operations and File Management</b>		
Q9.	Create and Write to a File: Write a Java program that prompts the user for a diary entry, then creates a file named " <b>diary.txt</b> " and writes the current date followed by the user's entry into this file. Ensure the program checks if the file already exists and informs the user, to avoid overwriting any previous content.	L1, L2
Q10.	Read from a File: Write a Java application that opens the " <b>diary.txt</b> " file created in the previous question and displays its content on the console. The program should handle cases where the file does not exist by displaying an appropriate error message.	L1, L2
Q11.	Append Content to an Existing File: Write a Java program that adds a new diary entry to the " <b>diary.txt</b> " file without overwriting its existing content. The program should ask the user for the new entry and append it to the file along with a timestamp.	L1, L2
Q12.	List Files and Directories: Write a program in Java that asks the user for a directory path and then lists all files and subdirectories in that directory. If the directory does not exist, the program should inform the user.	L2, L3
Q13.	Delete a Specific File: Write a Java program where the user can enter the name of a file to be deleted from the system. The program should check if the file exists and delete it, providing a confirmation message upon successful deletion or an error message if the file does not exist.	L1, L2
Q14.	Copy File Content: Write a Java program that copies the content from one file (source) to another (destination). The program should prompt the user for both source and destination file paths and perform the copy operation, ensuring that it doesn't overwrite	L3, L4

	an existing file without user confirmation.	
Q15.	Rename a File: Develop a Java application that renames a specified file. The program should request the current file name and the new file name from the user, renaming the file accordingly and confirming the action upon completion.	L2, L3
Q16.	Display File Metadata: Create a Java program that displays metadata of a specified file. The user should be able to input the file name, and the program should output the file size, last modified date, and other available attributes.	L2, L3
	<b>-END-</b>	