

Assignment-5 Text and Binary Files
Subject: Computer Science Workshop - 1 (CSE 2141)
Session: Sep 2025 to Jan 2026
Branch: Computer Science and Engineering (CSE)
Section: All
Course Outcome: CO1, CO2, CO3
Program Outcomes: PO1, PO2, PO3, and PO5
Learning Levels: Remembering (L1), Understanding (L2), Application (L3)

Q no.	Questions	Learning Levels
Q1.	<p>Write a Python program to create a text file named students.txt and store the names of five students in it. After writing, open the file in read mode and display its contents on the screen. Your program should ensure proper file handling (open, write, read, and close). Use the with keyword for safer file operations.</p> <p>Input:</p> <p style="padding-left: 40px;">["Alice", "Bob", "Charlie", "David", "Eva"]</p> <p>Output:</p> <p style="padding-left: 40px;">File 'students.txt' created successfully. Contents of the file: Alice Bob Charlie David Eva.</p>	L1, L2
Q2.	<p>Write a Python program to demonstrate file exception handling. The program should attempt to open a file named data.txt. If the file does not exist, display an error message like “Error: File not found.” instead of crashing. If the file exists, print its contents.</p> <p>Input:</p> <p style="padding-left: 40px;">Case 1: data.txt does not exist Case 2: data.txt contains → Python is powerful.</p> <p>Output:</p> <p style="padding-left: 40px;">Case 1: Error: File not found. Case 2: File opened successfully. Contents of data.txt: Python is powerful.</p>	L1, L2
Q3.	<p>Write a program that uses the with keyword to handle files safely. The program should open a file called poem.txt, write a short poem into it, and then automatically close the file without explicitly calling close(). After writing, reopen the file in read mode and print the poem line by line.</p> <p>Input:</p>	L1, L2

	<p>Roses are red, Violets are blue, Python is fun, And so are you.</p> <p>Output:</p> <p>Poem written to file successfully. Reading back from poem.txt: Roses are red, Violets are blue, Python is fun, And so are you.</p>	
Q4.	<p>Write a program to read a text file line by line and count:</p> <ul style="list-style-type: none"> • The total number of lines. • The total number of words. • The total number of characters. <p>Print these counts as the output.</p> <p>Input:</p> <p>Python is great. It makes coding simple.</p> <p>Output:</p> <p>Number of lines: 2 Number of words: 6 Number of characters: 38</p>	L1, L2
Q5.	<p>Write a Python program that demonstrates the use of a file pointer (seek()). The program should:</p> <ul style="list-style-type: none"> • Open a text file, • Read the first 10 characters, • Move the file pointer back to the beginning, • Read and print the entire file. <p>Input:</p> <p>Hello Python World!</p> <p>Output:</p> <p>First 10 characters: Hello Pyth Full file contents after seek(): Hello Python World!</p>	L2, L3
Q6.	<p>(a) Write a Python code that asks the user to enter a diary note and then attempts to create a text file named "diary.txt". Before writing, the program must check whether the file already exists; if it does, display a message informing the user that the file already exists and avoid overwriting any previous content. If the file does not exist, the program should create it and write the current date followed by the user's diary entry.</p> <p>(b) In a second part, write another Python code that attempts to open and read the contents of "diary.txt", displaying its full content on the screen; if the file is missing, the program should handle the error gracefully by printing an appropriate message such as <i>"File not found. Please check the name."</i>. Ensure both programs use proper file handling (including the with statement) and exception handling techniques.</p>	L2, L3

Q7.	<p>Write a Python program that performs direct binary read and write operations using the struct module. The program should save a list of integers into a binary file and then read them back to confirm correctness.</p> <p>Input:</p> <p style="padding-left: 40px;">[10, 20, 30, 40, 50]</p> <p>Output:</p> <p style="padding-left: 40px;">Integers written to binary file successfully. Integers read from file: [10, 20, 30, 40, 50].</p>	L2, L3
Q8.	<p>Write a Python program to compare storing fixed-length vs. variable-length strings in a file.</p> <ul style="list-style-type: none"> • Save a string in fixed length (20 characters padded with spaces). • Save another string in variable length. <p>Read both back and explain the storage difference.</p> <p>Input:</p> <p style="padding-left: 40px;">String: "Python"</p> <p>Output:</p> <p style="padding-left: 40px;">Fixed length storage: 'Python ' (20 bytes) Variable length storage: 'Python' (6 bytes).</p>	L2, L3
Q9.	<p>Write a program to demonstrate the difference between Big Endian and Little Endian byte ordering using the struct module. Store the same integer in both formats and display the raw byte sequences.</p> <p>Input:</p> <p style="padding-left: 40px;">1025</p> <p>Output:</p> <p style="padding-left: 40px;">Little Endian: b'\x01\x04' Big Endian: b'\x04\x01'.</p>	L2, L3
Q10.	<p>Write a Python program that uses the Pickling module.</p> <ul style="list-style-type: none"> • Pickle a dictionary containing student records (name, age, grade) into a binary file. • Unpickle the data and display it. <p>Input:</p> <p style="padding-left: 40px;">{"Alice": (20, "A"), "Bob": (19, "B"), "Charlie": (21, "A")}</p> <p>Output:</p> <p style="padding-left: 40px;">Pickled data saved successfully. Unpickled data: {'Alice': (20, 'A'), 'Bob': (19, 'B'), 'Charlie': (21, 'A')}</p>	L2, L3
	-END-	