

## Assignment 2 for Advanced List Capabilities

Subject: Computer Science Workshop - 1 (CSE 2141)

Session: September 2025 to January 2026

Branch: Computer Science and Engineering (CSE)

Section: All

Course Outcomes: CO1, CO2

Program Outcomes: PO1, PO2, and PO5

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

Q no.	Questions	Learning Levels
Q1.	<p>Write a Python program that stores the scores of students in a list and performs various operations on it. The program should accomplish the following tasks:</p> <ol style="list-style-type: none"><li>Find and display the <b>average score</b>.</li><li>Determine the <b>minimum</b> and <b>maximum</b> scores.</li><li>Display all <b>scores above the average</b>.</li><li><b>Sort</b> the list in <b>descending order</b> of scores.</li><li><b>Replace the three lowest scores</b> in the list with zero (0).</li></ol>	L1, L2
Q2.	<p>Write a Python function that classifies integers in a given list according to a custom classification scheme. The function should divide numbers into meaningful categories and return the result as a <b>dictionary</b>, where the keys are category names and the values are lists of numbers belonging to those categories.</p> <p>Possible classification schemes include:</p> <ul style="list-style-type: none"><li>• <b>Prime Numbers</b></li><li>• <b>Composite Numbers</b></li><li>• <b>Perfect Squares</b></li><li>• <b>Perfect Cubes</b></li></ul> <p>The function should process the input list, identify numbers belonging to each category, and display the resulting dictionary in a clear format.</p> <p><b>Input:</b> A list of integers entered by the user (e.g., [2, 4, 8, 9, 27, 28]).</p> <p><b>Output:</b> Display a dictionary showing the classification of numbers under different categories.</p> <p><b>Example Output:</b></p> <pre>Input = [2, 4, 8, 9, 27, 28] Output = [     "Prime": [2],     "Composite": [4, 8, 9, 27, 28],     "Perfect Squares": [4, 9],     "Perfect Cubes": [8, 27] ]</pre>	L2, L3

Q no.	Questions	Learning Levels
Q3.	<p>Write a menu-driven Python program to implement a <b>Stack</b> using a list. Define separate methods for each stack operation as follows:</p> <ul style="list-style-type: none"> <li>• <code>push(x)</code> – To insert an element into the stack.</li> <li>• <code>pop()</code> – To remove and return the top element from the stack.</li> <li>• <code>is_empty()</code> – To check whether the stack is empty.</li> <li>• <code>display()</code> – To display all elements of the stack.</li> </ul> <p>After implementing the stack operations, write a separate method to <b>evaluate a Reverse Polish Notation (RPN)</b> expression using the stack. The RPN evaluation should correctly handle operands and operators, following postfix evaluation rules.</p> <p><b>Example Output:</b>  Stack Operations:  1. Push  2. Pop  3. Display  4. Evaluate RPN  Enter choice: 4  Enter RPN Expression: 5 3 4 * +  Result: 17</p>	L2, L3
Q4.	<p>Write a Python program to perform various matrix operations using <b>nested lists</b>. The program should perform the following tasks:</p> <ol style="list-style-type: none"> <li>Create two <math>2 \times 2</math> matrices using nested lists.</li> <li>Compute and print their <b>sum</b> and <b>product</b> (matrix multiplication).</li> <li>Display both matrices and their results in a proper formatted output.</li> <li>Sort the rows of the resultant matrix based on the <b>sum of each row</b> and display the sorted matrix.</li> </ol> <p><b>Input:</b> Elements of two <math>2 \times 2</math> matrices entered by the user.  <b>Output:</b> Display the two input matrices, their sum, product, and the sorted resultant matrix (based on row sums).</p>	L2, L3
Q5.	<p>Write a Python function that takes a dictionary containing <b>student names</b> as keys and their <b>list of scores</b> (integers) as values. The function should calculate the average score for each student and return the name of the student who has the <b>highest average score</b>.</p> <p><b>Input:</b> A dictionary where each key is a student's name and each value is a list of integer scores. (e.g., <code>students_scores = {"Ram": [85, 90, 92], "Laxman": [70, 80, 88], "Janaki": [95, 100, 90]}</code>)  <b>Output:</b> Display the name of the student with the highest average score.</p>	L2, L3
Q6.	<p>Write a Python function that takes a dictionary where:</p> <ul style="list-style-type: none"> <li>• The <b>values</b> can be either a <b>list</b> or a <b>tuple</b> of integers.</li> <li>• If the value is a <b>list</b>, find the <b>sum of all prime numbers</b> in that list.</li> <li>• If the value is a <b>tuple</b>, find the <b>product of all odd numbers</b> in that tuple.</li> </ul> <p>The function should return a new dictionary where the keys remain the same, but the values are replaced with the computed results according to the above rules.</p> <p><b>Input:</b> A dictionary containing lists or tuples of integers. (e.g., <code>data = {"A": [2, 3, 4, 5, 10], "B": (1, 2, 3, 4, 5), "C": [7, 8, 9], "D": (6, 7, 8)}</code>)  <b>Output:</b> Display the new dictionary after computing the respective operations.</p>	L2, L3

Q no.	Questions	Learning Levels
Q7.	<p>Write a Python program that demonstrates the use of <b>set comprehension</b>, <b>dictionary comprehension</b>, and <b>lambda function</b> with sorting. The program should perform the following tasks:</p> <ol style="list-style-type: none"> <li>Accept a list of integers with duplicate values.</li> <li>Remove duplicates using <b>set comprehension</b>.</li> <li>Construct a <b>frequency dictionary</b> using <b>dictionary comprehension</b>.</li> <li>Sort the numbers in <b>descending order of frequency</b> using the <code>sorted()</code> function with a lambda expression.</li> </ol> <p><b>Input:</b> A list of integers with duplicates (e.g., [4, 2, 7, 4, 2, 4, 9, 7, 9, 9]).</p> <p><b>Output:</b> Display the unique numbers, frequency dictionary, and numbers sorted list by descending frequency.</p>	L1, L2
Q8.	<p><b>Description:</b> Write a Python program to process a list of email-like strings using <b>functional programming tools</b>. The program should perform the following tasks:</p> <ol style="list-style-type: none"> <li>Use <code>filter()</code> with a lambda function to extract only <b>valid emails</b> (those that contain @ and end with .com or .org).</li> <li>Use <b>list comprehension</b> to extract the <b>domain names</b> from the valid emails.</li> <li>Construct a <b>dictionary</b> where each key is a domain name and each value is its frequency count.</li> </ol> <p><b>Input:</b> A list of email-like strings (e.g., ["test@gmail.com", "hello123", "abc.org", "world@yahoo.com"]).</p> <p><b>Example Output:</b>  Valid Emails: ["test@gmail.com", "world@yahoo.com"]  Domains: ["gmail", "yahoo"]  Domain Frequency: "gmail": 1, "yahoo": 1</p>	L2, L3
Q9.	<p>Write a Python program that checks <b>voting eligibility</b> using <code>filter()</code> and <b>dictionary comprehension</b>. The program should perform the following tasks:</p> <ol style="list-style-type: none"> <li>Use <code>filter()</code> to extract <b>eligible voters</b> (age <math>\geq 18</math> and nationality = "Indian").</li> <li>Count and display the total number of eligible voters.</li> <li>Build a dictionary using <b>dictionary comprehension</b> in the format {'Eligible': [...], 'Not Eligible': [...]}.</li> </ol> <p><b>Input:</b> A list of tuples containing voter information — name, age, and nationality. (e.g., [("Amit", 22, "Indian"), ("John", 30, "USA"), ("Neha", 17, "Indian"), ("Ravi", 19, "Indian")])</p> <p><b>Output:</b> Eligible: ["Amit", "Ravi"]  Count: 2  {'Eligible': ['Amit', 'Ravi'], 'Not Eligible': ['John', 'Neha']}</p>	L2, L3

Q no.	Questions	Learning Levels
Q10.	<p>The student information is stored using a combination of different Python data structures:</p> <ul style="list-style-type: none"> <li>• <b>List</b> – contains marks of students in different subjects.</li> <li>• <b>Tuple</b> – stores a student's information such as (ID, Name).</li> <li>• <b>Set</b> – keeps track of unique skills a student has (e.g., "Python", "Java", "C", "C#", "C++").</li> <li>• <b>Dictionary</b> – represents a full student profile with keys like "info", "marks", and "skills".</li> </ul> <p>You are given a list of such student profiles (each represented as a dictionary). Write a Python function that processes this data and performs the following tasks:</p> <ul style="list-style-type: none"> <li>• Calculate and display the <b>average marks</b> of each student.</li> <li>• Count the total occurrences of each <b>skill</b> across all students (e.g., how many students know Python, Java, etc.).</li> <li>• Identify and display the <b>top-performing student</b> based on average marks.</li> </ul> <p><b>Input:</b> A list of student profiles represented as dictionaries, each containing "info", "marks", and "skills".</p> <p><b>Output:</b> Display the average marks of each student, the overall skill frequency, and the name of the top-performing student.</p>	L2, L3