THE SUPERIOR UNIVERSITY LAHORE

Assignment - 4

Semester: 3rd **Section: All** **Session: Fall 2025**

Faculty of Computer Science and Information Technology **QCH:**

Subject: Data Structures and Algorithms **Total Marks: 10**

Name.....

Roll No.....

Instructions:

Write the solution of the given question in hard form and submit the Scan PDF to LMS before the deadline.

Task #	CLO #	Domain and BT Level	Total Points
1	4	C3	10

Social Network Friend Recommendation System **(10 Marks)**

Scenario:
You are designing a module for a simplified social media platform. Each user is represented as a vertex in an undirected graph, and each friendship represents a bidirectional edge. Since the social network is large and connections are sparse, the platform stores all relationships using an Adjacency List representation.

Since the use friendship graph is given as:

User	Friends
A	B, C
B	A, C, D
C	A, B, D, F
D	B, C, F, H
F	C, D, H
H	D, F

1- **Add New User (E)**

Add a new user **E** to the graph and create the following friendships:

- E — C
- E — F

2- **Delete Friendship**

Remove the mutual (bidirectional) friendship between the following pair:

- C — D

3- Compute Degree of C

After performing the deletion in Step 2, compute and print the **degree** of vertex **C**.

4- Display Updated Adjacency List

Print the **complete updated adjacency list** for all vertices in the graph after all modifications.

5- Connectivity Check

Determine whether the updated graph remains:

- **Connected** (all vertices reachable from any other), **OR**
- **Disconnected** (graph splits into multiple components)

Use any graph traversal algorithm—BFS or DFS.

6- Draw the Updated Graph

Using a graph-drawing library such as **networkx + matplotlib**, generate a visual diagram of the final graph.
Your drawing must clearly show:

- All vertices (A, B, C, D, F, H, E)
- All edges (friendships)
- Inclusion of new edges (E—C, E—F)
- Removal of edge (C—D)

The graph should be neat, readable, and properly labeled.

7- Python Implementation

Write and execute a complete Python program that:

a. Implements an Adjacency List Graph Class

Includes methods for:

- Adding vertices
- Adding edges
- Removing edges
- Displaying adjacency list
- Computing degree
- Checking connectivity

b. Performs All Required Operations

Your code must sequentially execute:

- Add user **E** and its edges
- Delete friendship C—D
- Compute degree of C

- Display updated adjacency list
- Check connectivity
- Draw the updated graph

c. Shows Output to User

Your program must print:

- Final adjacency list
- Degree of C
- Connectivity result
- Display the drawn graph

Best of Luck Students!