



## National University of Computer & Emerging Sciences



**AL2002 – Artificial Intelligence – Lab (Spring 2025)**

**BSCS-6B**

### Lab Work 2 (Advanced Python)

Lab Instructor	Momna Javaid
Department	Computer Science



## Instructions:

### Note: Carefully read the following instructions

1. You also have to submit .ipynb file.
2. Plagiarism is strictly prohibited, 0 marks would be given to students who cheat.
3. Write codes in PYTHON language and you may use any IDE or Notebook environment.
4. First think about the problem statements before starting your programming.
5. At the end when you have done your tasks, attached .py or .ipynb files on google classroom.
6. Paste your complete code in word file along with output in case of .py file (Make sure your submission is completed). In case of missing any file, marks will be deducted.
7. Please submit your file in this format 22Fxxxx\_Name\_SecB\_Lab#
8. Do not submit your assignment after deadline. Late and email submission is not accepted.
9. Do not copy code from any source otherwise you will be penalized with ZERO Marks.
10. YOUR MARKING WILL BE BASED ON THE PRIOR SUBMISSION OF YOUR CODE BEFORE DEADLINE.

## Lab Tasks:

### Task 1:

A **smart irrigation system** named "**AquaFlow**" is used to efficiently water rectangular fields. To determine the water requirements, the system must calculate the **area** and **volume** of the field. Create a class named "**AquaFlow**", where:

- The user provides the **length**, **width**, and **depth** of the field.
- A **member function Calculate\_area()** computes the area of the field.
- A **member function Calculate\_volume()** computes the water volume required if the field is to be irrigated up to a given depth.
- A **member function display()** shows the calculated area and volume.
- At the end, the program prints: "**Irrigation Starts...**"

Note: Use tuple for input values.

### Task 2:

A **library** categorizes books into different **genres** to manage its collection efficiently. Perform various operations using built-in set functions.

Create a **class named "Library"**, which includes:

- Four sets representing different book genres: **Fiction, Non-Fiction, Science, History**
- A **member function to add a book** to a specific genre.
- A **member function to remove a book** from a genre.
- A **member function to check if a book exists** in a particular genre.



- **A member function to perform set operations:**
  - **Union** (books across multiple genres).
  - **Intersection** (books common in selected genres).
  - **Difference** (books that belong to one genre but not another).

Note: Use built-in functions for sets in this task.

### Task 3:

Create a system that asks the user about their car's condition and determines whether it needs maintenance. Use a dictionary to store predefined issue weights.

Signs that a car needs maintenance may include:

Engine noise: 4

Check engine light: 5

Poor fuel efficiency: 3

Strange vibrations: 3

Difficulty starting: 4

Braking issues: 5

Unusual exhaust smoke: 4

Steering problems: 3

Instructions:

Ask the user if they are experiencing each issue (Yes = 1, No = 0).

Sum the issue weights based on the user's input.

Define a threshold of 15 to determine if maintenance is needed.

Output Statements:

If the total score is above the threshold (15):

"Based on the issues reported, your car may require maintenance. It is recommended to visit a mechanic for a thorough inspection."

If the total score is below the threshold:

"Based on the issues reported, your car does not seem to require immediate maintenance.

However, if you are concerned, consulting a mechanic is always a good idea."

### Task 4:

A city wants to develop a navigation system that helps users find the shortest route between two locations. The city is represented as a graph, where intersections (nodes) are connected by roads (edges) with weights representing distances.

#### Graph Representation:

- **Nodes:** Locations
- **Edges:** Roads connecting locations, weighted by the distance between them.

#### Operations:

1. **Add locations and roads** to the city network.
2. **Display the city map**

Note: Use class concepts and graphs for writing this program.