**FAST School of Computing**

**Object Oriented Programming – Spring 2025**

**Software Engineering Department**

**LAB 08**

**Classes in C++**

**Learning Outcomes**

In this lab you are expected to learn the following:

Implement all Basic concepts of Classes, getter, setter and constructors

**Lab Activity 1:**

**FAST Sustainable Garden Management Case Study**

At FAST University’s community garden, the Garden Coordinator uses a simple system to manage everything smoothly. The coordinator can add new garden plots and store details about plants, like how much water and sunlight they need and how many fruits or vegetables they produce. The coordinator can Add Volunteers to the system and assigned to specific plots to carry out tasks such as watering, weeding, and harvesting and Volunteers can update plant status and report problems like pests or diseases. The coordinator can request the tools or materials needed for Gardening. The system also helps the coordinator track which tasks are completed and which are still pending. It shows how many tasks each volunteer has finished, helping the coordinator monitor their performance. The system also gives updates on plant health, so the coordinator knows which plots need extra care. Everything, including plots, volunteers, plant conditions, task progress, and tool requests, is shown on one simple dashboard ask user inputs for each tasks. This helps the coordinator make sure the garden runs well, the volunteers stay active, and the plants stay healthy.

**How to Identify Classes, Attributes and Functions from Scenario?**

**What are the main "things" (nouns) in the scenario?** These become **classes**.

**What characteristics (properties) do these things have?** These become **attributes**.

**What can these things "do" or what is done to them?** These become **functions/methods**.

**Problem 1:**

**Scenario-Based Case Study: Library Book Management**

The city library is creating a new system to manage books and help members easily. Each book will have details like title, author, pages, rating, genre, and if it’s available or borrowed. The system will label books with more than 300 pages as "Long Read" and books with a rating over 4.5 as "Highly Rated!" When a book is borrowed, it will show as unavailable and will change back to available when returned. If someone tries to borrow a book that’s already taken, the system will give a warning. Staff can update book ratings and details if needed. The system will also remind staff if a book is overdue after 30 days. Books will be grouped by genre, and users can search by title or author. It will track how often books are borrowed to help staff decide which books to restock. The system will also suggest other highly rated books from the same genre. If a book is old or damaged, staff can mark it as "Archived" so it can't be borrowed again. This system will make book management simple and help readers and staff.

**Problem 2:**

**Scenario-Based Case Study: Library Book Management**

A small engineering lab is building a simple system to manage both equipment and researchers using one class for both. Each unit (researcher or equipment) will have common details like ID, status (active/inactive), and usage data. If it’s a researcher, extra details will include training status, number of completed tasks, and the ID of equipment they are using. If it’s equipment, it will track how many times it’s been used, calculate average weekly usage, and mark itself for maintenance when usage crosses a set limit.

The system will:

* Assign equipment fairly by checking usage to avoid overuse of the same equipment.
* Let researchers update their training status after certification.
* Let equipment manage its usage stats on its own without showing who is using it.
* Let the coordinator view all researchers and equipment summaries.
* Mark researchers as "ready" or "not ready" based on training.
* Change equipment status to "available," "in use," or "under maintenance" automatically.
* Warn the coordinator if a researcher keeps using the same equipment too often.
* Alert when equipment needs an inspection after heavy use.
* Show a report of inactive researchers who haven’t completed tasks for a long time.

By using one simple class with extra details for researchers and equipment, the lab will improve how equipment is shared, track researcher progress, and keep equipment in good shape.

**Submission Details:**

1. Save each question .cpp file with your roll no and lab number e.g. i22-XXXX\_Lab8.cpp
2. Take screen shot of running test cases of tasks.
3. Zip the .cpp file and screen shots (Do not create .rar file) with roll no and lab no. e.g. i22-XXXX\_Lab8.zip.
4. Submit the zip file on google classroom.