

National University of Computer and Emerging Sciences



Lab Manual 08 Object Oriented Programming

| | |
|--------------------|--|
| Course Instructor | Mr. Usama Hassan |
| Lab Instructor (s) | Ms. Fariha Maqbool Mr. Sohaib Ahmad |
| Section | BSE-2C |
| Semester | Spring 2023 |

Department of Computer Science

Objectives

After performing this lab, students shall be able to:

- ✓ Understand and implement Composition, aggregation, and Association.

Exercise 1: Composition

Design a "Car" class that has a "Engine" class as a component. The "Car" class should have a method to start the engine, and the "Engine" class should have a method to display the engine status.

Instructions:

- Create a class "Engine" with a private attribute "status" and a public method "displayStatus()".
- Create a class "Car" that has a private attribute of type "Engine".
- Implement a method "startEngine()" in the "Car" class that sets the engine status and calls the "displayStatus()" method of the "Engine" class.

Exercise 2: Aggregation

Design a "Library" class that has a collection of "Book" objects. The "Library" class should allow adding books and displaying the details of all books in the library.

Instructions:

- Create a class "Book" with attributes "title", "author", and "publicationYear".
- Create a class "Library" that has a private attribute, a vector of type "Book".
- Implement methods "addBook()" and "displayBooks()" in the "Library" class.

Exercise 3: Association

Design a "Student" class and a "Course" class with a many-to-many association between them. Implement methods to enroll students in courses and display the list of students in a course or the list of courses a student is enrolled in.

Instructions:

- Create a class "Student" with attributes "name" and "studentId".
- Create a class "Course" with attributes "name" and "courseCode".
- Implement a many-to-many association between the "Student" and "Course" classes using vectors or other suitable containers.

- Implement methods "enrollStudent()" and "displayStudents()" in the "Course" class, and "enrollInCourse()" and "displayCourses()" in the "Student" class.

Exercise 4: Hospital Management System

Design a hospital management system that includes the following classes and their relationships:

Person: A base class that has attributes such as name, address, and email.

Patient: A derived class from Person, with additional attributes such as patientId and a list of appointments.

Doctor: A derived class from Person, with additional attributes such as employeeId and a list of appointments.

Appointment: A class with attributes such as appointmentId, appointmentDateTime, and a reference to the associated patient and doctor.

Department: A class with attributes such as departmentName and a list of doctors in the department.

Relationships:

Composition: A department has multiple doctors, and a doctor can only belong to one department.

Aggregation: A doctor has multiple appointments, but an appointment can exist independently of a doctor.

Association: An appointment has an association with a patient and a doctor who are involved in the appointment.

Basic C++ code structure with classes and function prototypes is attached for your reference.