Object Oriented Programming (EE), Spring 2024

Design Experiments (13 & 14)

University Management System [UMS]

LLO-2 (P3) PLO5

In these experiments you will **practice** building a university management system in C++ which will serve as a central hub for managing academic and administrative operations within an educational institute.

You will **try** to write a software which will do the following:

1. User Management:
   - The UMS includes robust user management functionality, allowing administrators to create and manage user accounts for students, faculty and administrative personnel.
   - Users can access the system using secure authentication mechanisms, such as usernames, passwords for security.
2. Course Management:
   - Course management features enable administrators and instructors to create and manage courses.
   - Course information, including course codes, names, credits, instructors and prerequisites, is stored and managed within the system.
3. Enrollment Management:
   - The UMS facilitates the enrollment process for students, allowing them to browse available courses. Administrators can monitor and manage enrollments, track student progress, and handle enrollment-related tasks efficiently.
4. Grade Management:
   - Grade management functionality enables instructors to record and manage student grades, assignments, exams, and assessments.
   - The system supports grading scales, grade calculation, and grade reporting for students.
5. Data Retrieval and Reporting:
   - The UMS includes powerful data retrieval and reporting tools that allow administrators and stakeholders to access, analyze, and generate reports on various aspects of academic and administrative data.

The system should encompass the following core functionalities: User Management Course Management Enrollment Management Grade Management Data Retrieval and Search

- Design and implement the necessary classes to represent users, courses, enrollments, and grades effectively.
- Develop methods for managing user data, course information, enrollment processes, and grading systems.
- Implement user-friendly interfaces or command-line interactions for users to interact with the university management system.

- Test and validate the functionality, reliability, and performance of the system through comprehensive testing procedures.

Experiment 13:

Under guidance you are to do the following:

1. Identify the objects that you will need. (CLASSES)
2. Identify the properties of the objects identified in part I above. (DATA MEMBERS)
3. Identify the functions (operations) that the objects of part I above will need. (MEMBER FUNCTIONS)
4. Description of how the following OOP principals have been used in your design.
    - Inheritance
    - Composition
    - Polymorphism
    - Encapsulation
    - Abstraction

Experiment 14:

Under guidance you are to use definitions worked in experiment 13 to continue with the following:

1. Write C++ Classes definitions (.h files) for all objects in Exp-13. All classes should match the objects in Exp-13.
2. Code C++ Classes for all objects (.cpp files) in activity 1.
3. Write a C++ program that implements the algorithm of the design phase.
4. Test your program with enough examples to demonstrate the working of your experiment.