***Database Management Systems (DBMS) & Object-Oriented Programming (OOP)***

***Project Report***

***Online Academy Management System***

***Submitted by:***

***YASHRA NOOR------(FA24B1-CS-047)***

***RAFIA MARYAM ------ (FA24B1-CS-060)***

***RAMESA SOHAIL------(FA24B1-CS-004)***

***FIZA TARIQ------(FA24B1-CS-017)***

***Submitted to:***

***Prof. Adeel Khalid***

***DATED:***

***11 JUNE,2025***

***The Fatima Jinnah Women University, Rawalpindi.***

# 1. Introduction

The Online Academy Management System is designed to simplify and automate the academic process for both students and instructors. Built using a combination of DBMS and OOP principles, the system ensures modularity, data integrity, and maintainability. It allows for efficient course management, enrollment, assignment handling, grading, and progress tracking.

# 2. System Architecture Overview

The system is divided into three main layers:  
- **Front-End (GUI):** Interface for students and instructors to interact with the system.  
- **Back-End (Java):** Handles the core logic such as course creation, student enrollment, assignment management, and grading.  
- **Database (MySQL):** Stores structured information about users, courses, enrollments, assignments, and grades.  
This layered architecture enables modular development, efficient troubleshooting, and future scalability.

# 3. Database Design

The system uses a relational schema for data integrity and normalization. The main tables include:

Students (student\_id, name, email)  
Courses (course\_id, name, description)  
Instructors (instructor\_id, name, email)  
Assignments (assignment\_id, course\_id, due\_date)  
Grades (grade\_id, enrollment\_id, score)  
Enrollments (enrollment\_id, student\_id, course\_id)

The schema is normalized to minimize redundancy and ensure efficient querying.

# 4. OOP Concepts Applied

**Encapsulation:** Each entity (e.g., Student, Course) is modeled as a class that bundles data and behavior.

**Inheritance**: Common functionality can be reused. For example, GraduateStudent may inherit from Student.

**Polymorphism:** Allows the system to handle different types of assignments or users through shared interfaces.

**Abstraction:** Only essential features are exposed; complexities are hidden within the class definitions.

# 5. Class Diagrams and Relationships

**Student Class**  
Attributes: studentID, name, email.  
Methods: enrollCourse, viewGrades.

**Instructor Class**  
Attributes: instructorID, name, email.  
Methods: createCourse, gradeAssignment.

**Enrollment Class**  
Attributes: enrollmentID, studentID, courseID.  
Methods: calculateGrade, dropCourse.

**Course Class**  
Attributes: courseID, name, description.  
Methods: addStudent, addAssignment.

Relationships between classes (e.g., one-to-many between Course and Assignments) are clearly defined, enabling a modular and maintainable codebase.

# 6. Functional Flow

**Student Workflow:**  
1. Login – Student accesses the system.  
2. Browse Courses – Student explores available options.  
3. Select & Enroll – Student selects and joins a course.  
4. Update Records – Enrollment is logged in both objects and database.

**Assignment and Grading Workflow:**  
1. Instructor Creates – Assignment details are added.  
2. Student Submits – File upload is completed.  
3. System Stores – Submission and status are updated.  
4. Instructor Grades – Instructor reviews and scores.  
5. System Notifies – Grades are stored; student is notified.

# 7. Technology Stack

**Database:** MySQL  
**Backend:** Java  
**Frontend:** GUI (Java Swing / AWT or other)

# 8. Benefits of DBS and OOP Integration

- Ensures data consistency and integrity via relational models.

- Promotes code reusability, modularity, and scalability via OOP.

- System is user-friendly, efficient, and future-proof.

# 9. Conclusion

The Online Academy Management System demonstrates the powerful synergy of DBMS and OOP. By combining structured data storage with modular programming techniques, the system streamlines educational workflows, improves academic communication, and enhances overall user experience. It stands as a scalable, efficient, and maintainable platform for digital learning environments.

# 10. RELATIONAL MODEL



















