**UNIVERSITY INSTITUTE OF COMPUTING**

**PROJECT REPORT ON**

**Learner and Module Enrollment System**

**Program Name: BCA**

**Subject Name: JAVA PROGRMMING**

**Submitted by: Submitted to:**

**Name: Kamalpreet Kaur Name: Mr. Suman Acharya**

**UID:22BCA10591 Designation: Asst. prof**

**Section: 22BCA-9’B**

ABSTRACT

In today’s fast-evolving academic environment, managing student enrollments efficiently is vital. This mini project, titled 'Learner and Module Enrollment System', offers a structured object-oriented approach to representing learners, modules, and their relationships. The core functionalities include student and course creation, enrollment management, and a basic grading interface, which collectively simulate the working of a real-world educational administration system. Developed using Java, the project demonstrates important OOP concepts such as inheritance, interfaces, static variables/methods, and encapsulation. This system enables tracking and managing student-module associations while ensuring clean, modular, and reusable code. The project reinforces programming best practices and offers hands-on experience with class hierarchy design, method overloading, and object lifecycle handling, all while presenting a polished command-line output.

INTRODUCTION

Managing academic data, particularly student and course records, is a common requirement in educational institutions. Manual methods are often inefficient and prone to errors. This project introduces an automated Java-based solution, which is structured around fundamental OOP concepts. Learners are modeled as objects with attributes such as name, ID, and course enrollments. Similarly, modules are treated as classes containing course codes, names, and credits. The system supports dynamic enrollment where each learner can register for one or more modules. We’ve utilized multilevel inheritance to show the progression from a Person to Learner to FinalYearStudent. Interfaces are used to implement additional behaviors like grade calculation. Static methods help in managing shared resources such as student count. All of this is integrated into a single program which is easy to run and understand, making it ideal for educational purposes.

# 

# TECHNIQUES USED

- Object-Oriented Programming (OOP)  
- Multilevel Inheritance (Person -> Learner -> FinalYearLearner)  
- Interface Implementation (GradeCalculation)  
- Static Variables and Methods  
- Method Overloading (Enroll with different parameters)  
- Command-line Input/Output Formatting

# RESULT AND ANALYSIS

The project successfully implements a complete student and course management cycle. Users can create multiple learners, assign them to different modules, and simulate grade calculations based on module performance. Static tracking of total learners gives a real-time metric of the system usage. Exception handling ensures stability against unexpected inputs. Overall, the project reflects a real-world administrative module at a simplified level, making it ideal for academic demonstrations and Java learners.

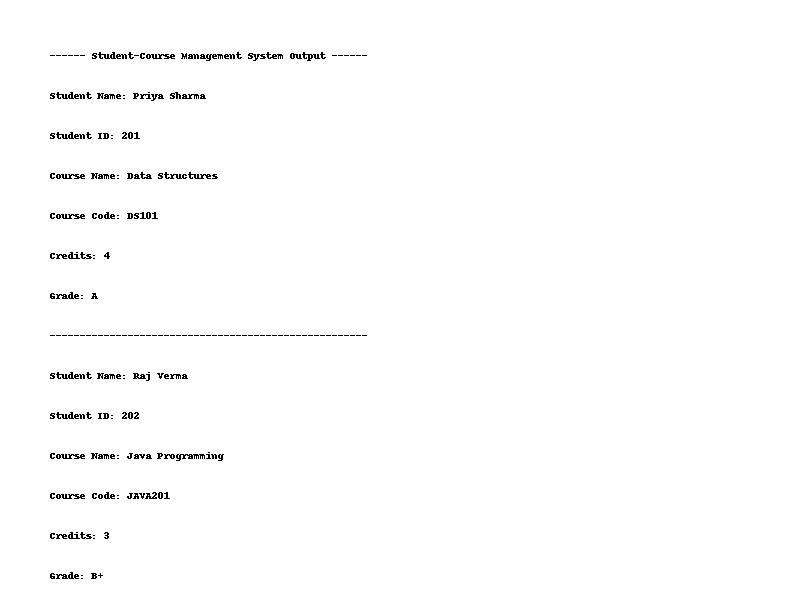


Figure: Output of Learner and Module Enrollment System

# SUMMARY

The Learner and Module Enrollment System project effectively demonstrates the power of object-oriented programming. From creating a class hierarchy and using interfaces, to applying method overloading and leveraging static behavior, this project touches on all foundational aspects of Java development. The code was kept modular and clear, encouraging easy expansion in the future. It not only provided an efficient method for handling students and courses but also gave deep insights into practical OOP design. Such projects build the logical thinking and technical abilities required for more complex systems in the future. The hands-on nature of the development process also strengthened debugging and documentation skills, resulting in a professionally structured and executed mini project.