CS 236: Data Structures and Algorithms –

Final Lab Assignment #5

Lab Assignment: University Course Registration System

Objective

The goal of this lab is to implement a simple **University Course Registration System** using **Object-Oriented Programming (OOP)** principles, **file handling**, and **data structures** in Python. Students will practice designing a system that allows students to enroll in courses, manage their schedules, and store data persistently.

Requirements

1. Classes to Implement

- **`Student`**: Represents a student with attributes like student ID, name, and registered courses.
- **`Course`**: Represents a course with attributes like course ID, name, instructor, and enrolled students.
- **`EnrollmentSystem`**: Manages student-course registrations and maintains records.

2. Data Structures to Use

- **Dictionary**: Store student and course information (`student_id: Student` and `course_id: Course`).
- **Lists/Sets**: Manage enrolled students for each course and registered courses for students.

3. Functional Requirements

✔ Allow students to **register** for an account.

✔ Allow students to **view available courses**.

✔ Enable students to **enroll** in a course (prevent schedule conflicts).

✔ Allow students to **drop courses**.

✔ Store enrollment records in a **CSV file**.

✔ Implement **error handling** for invalid inputs. ✔ Ensure a course does not exceed **maximum capacity** (e.g., 30 students per course).

Implementation Details

1. Student Class

```
class Student:
    def __init__(self, student_id, name):
        self.student_id = student_id
        self.name = name
        self.registered_courses = set()
```

2. Course Class

```
class Course:
    def __init__(self, course_id, name, instructor, max_students=30):
        self.course_id = course_id
        self.name = name
        self.instructor = instructor
        self.enrolled_students = set()
        self.max_students = max_students
```

3. Enrollment System Class

```
import csv
class EnrollmentSystem:
    def __init__(self):
        self.students = {}
        self.courses = {}
        self.load_data()

    def load_data(self):
        """Load student and course data from CSV file."""
        pass # Implement file reading logic here

    def enroll_student(self, student_id, course_id):
        """Enroll a student in a course."""
        pass # Implement enrollment logic
```

CSV File Handling

- **`students.csv`**: Stores student information (`student_id, name, registered_courses`).
- **`courses.csv`**: Stores course details (`course_id, name, instructor, enrolled_students`).
- **`enrollments.csv`**: Stores all enrollments (`student_id, course_id`).

Example Usage

```
enrollment_system = EnrollmentSystem()
enrollment_system.enroll_student("S101", "CS101")
```

```
enrollment_system.drop_course("S101", "CS101")
enrollment_system.view_available_courses()
```

Grading Criteria

- **Correct implementation of classes (10%)**
- **Efficient use of data structures (10%)**
- **Proper file handling (10%)**
- **Error handling and validation (10%)**
- **User-friendly interaction (10%)**
- **Presentation (The ability to answer any question and edit/ update your code promptly) (50%)**

Submission Guidelines and Deliverables:

- Submit your Python script (`.py` file)
- **Include a short document explaining:**
  - The role of each group member.
  - Challenges faced and how they were resolved.
- Ensure your code is well-documented and follows Python best practices.

Good luck, and happy coding!