

Exercise Description:

You are required to develop a Java program that models a student course registration system within a department. The program will involve creating students, courses, classrooms, and a registry that manages all courses offered in the department.

Details:

1. Student Class (Student.java):

- Represents a student with attributes:
 - `name`: The student's name.
 - `id`: The student's ID.
 - `courses`: A list of courses the student is enrolled in.
- **Methods:**
 - `printInfo()`: Prints the student's name and ID.
 - `addCourse(Course aCourse)`: Attempts to enroll the student in a course. If the course is not full, the student is enrolled.
- **Constructor:**
 - Initializes the student's name, ID, and course list.

2. Course Class (Course.java):

- Represents a course with attributes:
 - `name`: The name of the course.
 - `ects`: The number of ECTS credits for the course.
 - `room`: The classroom assigned to the course.
 - `students`: A list of students enrolled in the course.
- **Methods:**
 - `setClassroom(Classroom aRoom)`: Sets the classroom for the course.
 - `enroll(Student aStudent)`: Enrolls a student in the course.
 - `printCourseDetails()`: Prints course details, including the students enrolled.
 - `isFull()`: Checks whether the course has exceeded the capacity of the classroom.
- **Constructor:**
 - Initializes the course with a name and ECTS credits.

3. Classroom Class (Classroom.java):

- Represents a classroom with attributes:
 - `location`: The location of the classroom.
 - `capacity`: The maximum number of students the classroom can hold.
- **Methods:**
 - `getCapacity()`: Returns the capacity of the classroom.
 - `printLocation()`: Prints the location of the classroom.
- **Constructor:**
 - Initializes the classroom with a location and capacity.

4. Registry Class (Registry.java):

- Represents a department's course registry with attributes:
 - `departmentName`: The name of the department.
 - `allCourses`: A list of courses offered by the department.
- **Methods:**
 - `addCourse(Course aCourse)`: Adds a course to the registry.
 - `printAllRecords()`: Prints all courses and their details (including the enrolled students).
- **Constructor:**
 - Initializes the registry with the department name.

5. Main Class (Main.java):

- The main class creates instances of students, courses, classrooms, and the registry, demonstrating the interaction between these objects.
- **Process:**
 - Creates students and courses.
 - Assigns classrooms to courses.
 - Enrolls students in courses.
 - Prints the course and student details using the registry.

Task:

1. Implement the `Student`, `Course`, `Classroom`, and `Registry` classes as described.
2. Ensure that students can enroll in courses and that the system checks whether the course is full before enrolling a student.
3. In the `Main` class, simulate the creation of a department registry, courses, classrooms, and student enrollments.
4. Print the final list of all courses and their enrolled students.

Bonus:

- Add functionality to allow students to drop courses.
- Add validation to prevent students from enrolling in the same course multiple times.
- Allow the classroom capacity to dynamically adjust, e.g., add or remove seats.

This exercise focuses on object-oriented principles such as encapsulation, aggregation, and interaction between multiple classes.