Exercise Description:

You are required to develop a Java program that models the relationship between students and courses. The program will involve creating students and enrolling them in multiple courses, with a maximum limit of courses a student can take.

Details:

1. **Student Class (Student.java)**:
    - Represents a student with attributes:
        - `name`: The name of the student.
        - `id`: The student's ID.
        - `courses[]`: An array of `Course` objects representing the courses the student is enrolled in, with a maximum of 8 courses.
        - `numberOfCourses`: Tracks how many courses the student is currently enrolled in.
    - **Methods**:
        - `printInfo()`: Prints the student's name, ID, and details of all the courses they are enrolled in.
        - `addCourse(Course aCourse)`: Enrolls the student in a course if they have not already enrolled in 8 courses.
    - **Constructor**:
        - Initializes the student's name, ID, and the array for courses.
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.
        - `myStudent`: A reference to the student enrolled in the course (used for future expansions if needed).
    - **Methods**:
        - `setStudent(Student myStudent)`: Associates a student with the course.
        - `printCourseDetails()`: Prints the course name and ECTS credits.
    - **Constructor**:
        - Initializes the course with its name and ECTS credits.
3. **Main Class (Main.java)**:
    - The main class demonstrates the interaction between students and courses.
    - **Process**:
        - A student is created with a name and ID.
        - The student is enrolled in two courses, "Java" and "Machine Learning."
        - The student's information and the courses they are enrolled in are printed.

Task:

1. Implement the `Student` class with methods for printing student information and enrolling them in courses.
2. Implement the `Course` class with methods for printing course details and associating a student with the course.
3. In the `Main` class, simulate the creation of students and courses, enroll the students in courses, and print the final details.

Bonus:

- Add functionality to drop courses if needed.
- Expand the program to handle additional attributes for courses or students, such as grades.

This exercise covers basic object-oriented programming principles such as class relationships, arrays, and methods for interacting between objects in Java.