Laboratory Activity # 5: Modeling a School Management System

```
// TeacherStudentAssignment.java

import java.util.*;

// ======================
// Person Class
// ======================
class Person {
    protected String name;
    protected int age;
    protected String gender;

    public Person(String name, int age, String gender) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }

    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Gender: " + gender);
    }
}

// ======================
// Student Class
// ======================
class Student extends Person {
    private String studentId;

    public Student(String name, int age, String gender, String studentId) {
        super(name, age, gender);
        this.studentId = studentId;
    }

    public void displayStudent() {
        System.out.println("--- Student Information ---");
        displayInfo();
        System.out.println("Student ID: " + studentId);
        System.out.println();
    }
```

```
}

// ======================
// Course Class
// ======================
class Course {
    private String courseCode;
    private String courseName;

    public Course(String courseCode, String courseName) {
        this.courseCode = courseCode;
        this.courseName = courseName;
    }

    public void displayCourse() {
        System.out.println(courseCode + " - " + courseName);
    }

    public String getCourseCode() {
        return courseCode;
    }

    public String getCourseName() {
        return courseName;
    }
}

// ======================
// Teacher Class
// ======================
class Teacher extends Person {
    private String department;
    private List<Course> courses; // aggregation (HAS-A relationship)

    public Teacher(String name, int age, String gender, String department) {
        super(name, age, gender);
        this.department = department;
        this.courses = new ArrayList<>();
    }

    public void addCourse(Course course) {
        courses.add(course);
    }
```

```java
    public void displayTeacher() {
        System.out.println("--- Teacher Information ---");
        displayInfo();
        System.out.println("Department: " + department);
        System.out.println("Courses Handled:");
        for (Course c : courses) {
            c.displayCourse();
        }
        System.out.println();
    }
}

// ======================
// Main Class
// ======================
public class TeacherStudentAssignment {
    public static void main(String[] args) {

        // Create a Student with your details
        Student student = new Student("Airon Ursua", 20, "Male", "202411217");
        student.displayStudent();

        // Create Courses
        Course course1 = new Course("CS101", "Programming Fundamentals");
        Course course2 = new Course("CS102", "Object-Oriented Programming");

        // Create a Teacher and add courses
        Teacher teacher = new Teacher("Mr. Dela Cruz", 45, "Male", "Computer Studies");
        teacher.addCourse(course1);
        teacher.addCourse(course2);
        teacher.displayTeacher();

        // Display a single course
        System.out.println("Displaying the course");
        course1.displayCourse();
    }
}
```

```
--- Student Information ---
Name: Airon Ursua
Age: 20
Gender: Male
Student ID: 202411217
```

```
--- Teacher Information ---
Name: Mr. Dela Cruz
Age: 45
Gender: Male
Department: Computer Studies
Courses Handled:
CS101 - Programming Fundamentals
CS102 - Object-Oriented Programming

Displaying the course
CS101 - Programming Fundamentals
```

https://chatgpt.com/share/68fc39a2-8c00-8005-b04f-850349ac7a45