■ Extended Campus Course & Records Manager (CCRM)

■ Introduction

This extended version manages:
- ■ Student academic records
- ■ Course registration & scheduling
- ■ Faculty assignments
- ■ Attendance tracking
- ■ Grading system

■ Getting Started

```
javac src/*.java -d out
java -cp out Main
```

■ Core Classes

Student.java

```
import java.util.HashMap;

public class Student {
    private int id;
    private String name;
    private String[] courses;
    private HashMap<String, Integer> attendance;  // Course → Attendance %
    private HashMap<String, String> grades;       // Course → Grade

    public Student(int id, String name, String[] courses) {
        this.id = id;
        this.name = name;
        this.courses = courses;
        this.attendance = new HashMap<>();
        this.grades = new HashMap<>();

        // Initialize attendance and grades
        for (String c : courses) {
            attendance.put(c, 0);
            grades.put(c, "N/A");
        }
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public String[] getCourses() { return courses; }

    public void addCourse(String course) {
        String[] updated = new String[courses.length + 1];
        System.arraycopy(courses, 0, updated, 0, courses.length);
        updated[courses.length] = course;
        this.courses = updated;
        attendance.put(course, 0);
        grades.put(course, "N/A");
    }

    public void updateAttendance(String course, int percent) {
        attendance.put(course, percent);
    }

    public void assignGrade(String course, String grade) {
        grades.put(course, grade);
    }

    @Override
    public String toString() {
        return id + " - " + name + " → " + String.join(", ", courses);
    }

    public void showReport() {
        System.out.println("\n■ Report for " + name);
        for (String course : courses) {
            System.out.println("  " + course +
                               " | Attendance: " + attendance.get(course) + "%" +
                               " | Grade: " + grades.get(course));
        }
    }
}
```

CourseManager.java

```
import java.util.ArrayList;

public class CourseManager {
    private ArrayList<Student> students = new ArrayList<>();

    public void addStudent(Student s) {
        students.add(s);
    }

    public void registerCourse(int studentId, String course) {
        for (Student s : students) {
            if (s.getId() == studentId) {
                s.addCourse(course);
                System.out.println("■ Course added for " + s.getName());
                return;
            }
        }
        System.out.println("■ Student not found.");
    }

    public void markAttendance(int studentId, String course, int percent) {
        for (Student s : students) {
            if (s.getId() == studentId) {
                s.updateAttendance(course, percent);
                System.out.println("■ Attendance updated for " + s.getName());
                return;
            }
        }
        System.out.println("■ Student not found.");
    }

    public void assignGrade(int studentId, String course, String grade) {
        for (Student s : students) {
            if (s.getId() == studentId) {
                s.assignGrade(course, grade);
                System.out.println("■ Grade assigned to " + s.getName());
                return;
            }
        }
        System.out.println("■ Student not found.");
    }

    public void listStudents() {
        for (Student s : students) {
            System.out.println(s);
        }
    }

    public void showReports() {
        for (Student s : students) {
            s.showReport();
        }
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
        CourseManager manager = new CourseManager();

        Student s1 = new Student(1, "Alice", new String[]{"Math101", "CS102"});
        Student s2 = new Student(2, "Bob", new String[]{"Eng201"});

        manager.addStudent(s1);
        manager.addStudent(s2);

        System.out.println("■ Current Students:");
        manager.listStudents();

        // Register new course
        manager.registerCourse(1, "Hist210");

        // Attendance
        manager.markAttendance(1, "Math101", 85);
        manager.markAttendance(2, "Eng201", 92);

        // Grades
        manager.assignGrade(1, "Math101", "A");
        manager.assignGrade(2, "Eng201", "B+");

        System.out.println("\n■ Final Reports:");
        manager.showReports();
    }
}
```

■■ Example Output

```
■ Current Students:
1 - Alice → Math101, CS102
2 - Bob → Eng201
■ Course added for Alice
■ Attendance updated for Alice
■ Attendance updated for Bob
■ Grade assigned to Alice
■ Grade assigned to Bob

■ Final Reports:

■ Report for Alice
  Math101 | Attendance: 85% | Grade: A
  CS102 | Attendance: 0% | Grade: N/A
  Hist210 | Attendance: 0% | Grade: N/A

■ Report for Bob
  Eng201 | Attendance: 92% | Grade: B+
```

■ Extended Repository Structure

```
■■■ src
■   ■■■ Main.java
■   ■■■ Student.java
■   ■■■ CourseManager.java
■■■ docs
■   ■■■ usage-guide.md
■■■ README.md
```