**package** intern;

**import** java.util.\*;

**class** Student {

**private** String name;

**private** **int** rollNumber;

**private** HashMap<String, Integer> courseRegistrations;

**private** HashMap<String, Integer> grades;

**public** Student(String name, **int** rollNumber) {

**this**.name = name;

**this**.rollNumber = rollNumber;

**this**.courseRegistrations = **new** HashMap<>();

**this**.grades = **new** HashMap<>();

}

**public** String getName() {

**return** name;

}

**public** **int** getRollNumber() {

**return** rollNumber;

}

**public** HashMap<String, Integer> getCourseRegistrations() {

**return** courseRegistrations;

}

**public** HashMap<String, Integer> getGrades() {

**return** grades;

}

**public** **void** registerCourse(String courseCode, **int** credits) {

courseRegistrations.put(courseCode, credits);

System.***out***.println("Course " + courseCode + " registered successfully.");

}

**public** **void** assignGrade(String courseCode, **int** grade) {

grades.put(courseCode, grade);

System.***out***.println("Grade assigned successfully for course " + courseCode);

}

}

**class** Transcript {

**private** Student student;

**private** HashMap<String, Integer> courseCredits;

**private** HashMap<String, Integer> courseGrades;

**public** Transcript(Student student) {

**this**.student = student;

**this**.courseCredits = **new** HashMap<>();

**this**.courseGrades = **new** HashMap<>();

}

**public** **void** generateTranscript() {

System.***out***.println("Transcript for " + student.getName() + " (Roll Number: " + student.getRollNumber() + "):");

**for** (Map.Entry<String, Integer> entry : courseCredits.entrySet()) {

String courseCode = entry.getKey();

**int** credits = entry.getValue();

**int** grade = courseGrades.getOrDefault(courseCode, -1);

System.***out***.println("Course: " + courseCode + ", Credits: " + credits + ", Grade: " + (grade == -1 ? "Not available" : grade));

}

}

**public** **void** addCourse(String courseCode, **int** credits) {

courseCredits.put(courseCode, credits);

}

**public** **void** addGrade(String courseCode, **int** grade) {

courseGrades.put(courseCode, grade);

}

}

**class** StudentInformationSystem {

**private** ArrayList<Student> students;

**private** ArrayList<Transcript> transcripts;

**public** StudentInformationSystem() {

**this**.students = **new** ArrayList<>();

**this**.transcripts = **new** ArrayList<>();

}

**public** **void** addStudent(String name, **int** rollNumber) {

Student student = **new** Student(name, rollNumber);

students.add(student);

System.***out***.println("Student " + name + " added successfully.");

}

**public** **void** registerCourse(**int** rollNumber, String courseCode, **int** credits) {

Student student = findStudentByRollNumber(rollNumber);

**if** (student != **null**) {

student.registerCourse(courseCode, credits);

} **else** {

System.***out***.println("Student with Roll Number " + rollNumber + " not found.");

}

}

**public** **void** assignGrade(**int** rollNumber, String courseCode, **int** grade) {

Student student = findStudentByRollNumber(rollNumber);

**if** (student != **null**) {

student.assignGrade(courseCode, grade);

} **else** {

System.***out***.println("Student with Roll Number " + rollNumber + " not found.");

}

}

**public** **void** generateTranscript(**int** rollNumber) {

Student student = findStudentByRollNumber(rollNumber);

**if** (student != **null**) {

Transcript transcript = **new** Transcript(student);

**for** (Map.Entry<String, Integer> entry : student.getCourseRegistrations().entrySet()) {

transcript.addCourse(entry.getKey(), entry.getValue());

}

**for** (Map.Entry<String, Integer> entry : student.getGrades().entrySet()) {

transcript.addGrade(entry.getKey(), entry.getValue());

}

transcripts.add(transcript);

transcript.generateTranscript();

} **else** {

System.***out***.println("Student with Roll Number " + rollNumber + " not found.");

}

}

**private** Student findStudentByRollNumber(**int** rollNumber) {

**for** (Student student : students) {

**if** (student.getRollNumber() == rollNumber) {

**return** student;

}

}

**return** **null**;

}

}

**public** **class** StudentInformationSystemDemo {

**public** **static** **void** main(String[] args) {

StudentInformationSystem sis = **new** StudentInformationSystem();

sis.addStudent("ABC", 101);

sis.addStudent("XYZ", 102);

sis.registerCourse(101, "CS101", 3);

sis.assignGrade(101, "CS101", 85);

sis.registerCourse(102, "Math201", 4);

sis.assignGrade(102, "Math201", 92);

sis.generateTranscript(101);

sis.generateTranscript(102);

}

}