**Student Management System**

**1. Overview**

The Student Management System is a Java-based console application that allows administrators to manage student records and examination results efficiently. It demonstrates object-oriented programming concepts such as encapsulation, abstraction, and modular design.

**2. Features**

- Add and manage student records

- Enter and compute examination details

- Display individual student profiles

- Generate subject-wise and overall performance reports

**3. Class Descriptions**

3.1 Student Class

Represents individual student data.

Attributes:

- `name`

- `rollNumber`

- `dob` (Date of Birth)

- `email`

- `studentId`

- `course`

Key Methods:

- `displayDetails()`: Prints all student information

---

3.2 Subject Class

Represents individual subject performance.

Attributes:

- `name`

- `marks`

- `grade`

- `percentage`

Key Methods:

- `calculateGradeAndPercentage()`: Automatically calculates letter grade and percentage from marks

- `displaySubjectDetails()`: Shows subject marks and grade breakdown

---

3.3 Examination Class

Handles a collection of `Subject` instances and overall result aggregation.

Attributes:

- `subjects` (List of `Subject`)

- `overallPercentage`

- `overallGrade`

Key Methods:

- `addSubject(Subject subject)`: Adds a subject and triggers recalculation

- `displayExamResults()`: Displays complete subject-wise and cumulative result summary

3.4 StudentManagementSystem Class

The main driver class for user interaction and system operations.

Data Structures:

- `ArrayList<Student>` for storing student records

- `HashMap<Integer, Examination>` to map roll numbers to exam data

Operations:

- Add new students

- Input and save examination scores

- View student and result records

- Exit and control menu flow

**4. Grading Criteria**

| Marks (%) | Grade |

| 90–100 | A+ |

| 80–89 | A |

| 70–79 | B+ |

| 60–69 | B |

| 50–59 | C |

| 40–49 | D |

| Below 40 | F |

**5. Input Validation**

- Ensures marks are between 0–100

- Validates student existence before adding or viewing exam data

**6. Suggestions for Improvement**

- Export reports to files (CSV/PDF)

- Add login and user role functionality

- Integrate with a real database

- Add user-friendly GUI using JavaFX or Swing