

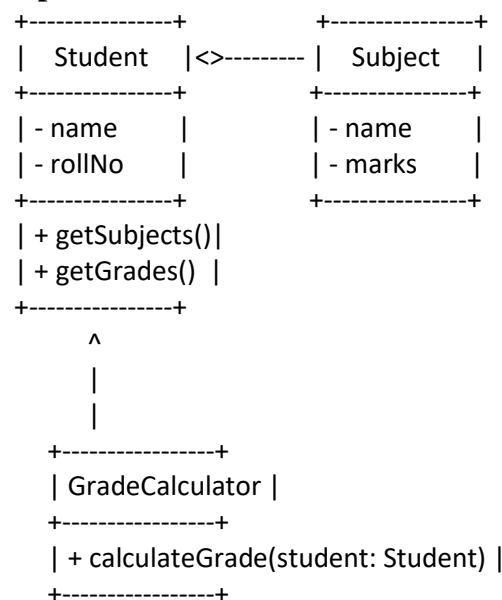
# School Results Application

## Class Diagram :-

Classes and Relationships:

- Student
    - Attributes: name, rollNo
    - Methods: getSubjects(), getGrades()
  - Subject
    - Attributes: name, marks
  - GradeCalculator
    - Methods: calculateGrade(Student student)
- Relationships:
- A Student aggregates multiple Subject objects → Aggregation (hollow diamond).
  - GradeCalculator uses Student (dependency).

## UML Representation:



## Object Diagram :-

Example Scenario:

- Student: John
- Subjects: Maths (90), Science (85)
- GradeCalculator computes result.

## UML Representation:

```
Student: John
-----
name = "John"
rollNo = 101
Subjects:
    Subject: Maths
        marks = 90
    Subject: Science
        marks = 85
```

```
GradeCalculator
-----
calculateGrade(John)
```

## **Sequence Diagram :-**

Scenario: Student requests grade.

- Actors: Student, GradeCalculator
- Flow:
  1. Student → GradeCalculator: `requestGrade()`
  2. GradeCalculator → Student: `getSubjects()`
  3. GradeCalculator calculates grade
  4. GradeCalculator → Student: `returnGrade()`

## **UML Representation:**

