Java Lab Assignment 1

Student Class Design & Basic Operations

Problem Statement

Create a Student Record Management system that allows the user to input, display, and calculate grades for students. Implement a class-based structure using Object-Oriented Programming principles to manage student data such as roll number, name, course, marks, and grade. The program should also allow the display of student records and calculate the grade based on marks.

Objective:
Introduce object-oriented concepts, control structures, input/output operations, and arrays/strings in Java.

Learning Outcomes

Upon completion of this assignment, the student will be able to:

1. Understand the fundamentals of object-oriented programming in Java.
2. Implement constructors, methods, and basic operations (input, output).
3. Work with arrays and strings in Java.
4. Use conditional statements and loops to control program flow.

Class Hierarchy & Data Types

Class Hierarchy:

- **Student** (inherits **Person**)
  - Fields: rollNo, name, course, marks, grade
  - Methods: inputDetails(), displayDetails(), calculateGrade()

Data Types:

- String: for name, course
- int: for rollNo
- double: for marks
- char: for grade

Detailed Instructions

1. **Core Design**: Create the Student class with fields like rollNo, name, course, marks, and grade.
2. **Constructors**: Implement a **default constructor** and **parameterized constructor** to initialize student details.
3. **Methods**:
    - inputDetails(): Take input from the user to add student details.
    - displayDetails(): Display student details.
    - calculateGrade(): Calculate the grade based on marks (A, B, C, D).
4. **Use Arrays**: Manage multiple student records using a **1D array** or **ArrayList**.

Expected Output

```
===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 1
Enter Roll No: 101
Enter Name: Rahul
Enter Course: B.Tech
Enter Marks: 87.0

===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 2
Roll No: 101
Name: Rahul
Course: B.Tech
Marks: 87.0
Grade: B
----------------------
Enter your choice: 3
Exiting the application. Goodbye!
```

Guidelines to Students

1. **Classes and Methods**: Follow proper class design principles. Define a Student class with methods to handle input, output, and grade calculation.

2. **Use of Arrays/Collections**: Manage multiple student records using an **ArrayList** or **1D array**.
3. **Menu Interaction**: Implement a simple text-based menu system using **Scanner** for user input.

Improvements/Adjustments

1. **Advanced Array Usage**: Use **2D arrays** or a **HashMap** for storing and managing multiple records.
2. **Data Validation**: Add validation for marks (i.e., ensure marks are between 0 and 100).

Submission Guidelines

1. Submit your **Java source code files**.
2. Ensure your program runs without errors. Provide a **README file** explaining how to run the code.

Performance Metrics (Out of 10 Marks)

Criteria	Marks
Core Design and Implementation	3
Array Handling and Data Validation	2
Methods Implementation	2
Menu System and User Interaction	2
Code Quality and Documentation	1