

Project Synopsis

Title: *Student Management System Using Java*

1. Introduction

The *Student Management System* is a console-based Java application designed to manage student records efficiently. It provides a structured and interactive way to handle key aspects of student data, including personal information, academic performance, and fee details. The system is developed using Object-Oriented Programming principles, making it modular, extensible, and easy to maintain.

2. Objective

The primary objective of this project is to create a simple yet functional management tool for educational institutions to:

- Maintain student records
 - Store and display examination marks and results
 - Manage and track fee payment details
 - Provide a consolidated view of a student's profile
-

3. Tools and Technology

- **Programming Language:** Java
- **Development Environment:** Any IDE or text editor (e.g., Eclipse, IntelliJ, or VS Code)
- **Core Concepts Used:**
 - Classes and Objects
 - Inheritance
 - Interfaces
 - Polymorphism
 - Collections (ArrayList)
 - Scanner for input

4. System Modules

The system is divided into the following main modules:

a. Student Module

- Captures student ID, name, course, and semester.
- Implemented through the Student class which implements the studentinterface.

b. Exam Module

- Accepts marks for five subjects.
- Calculates total, percentage, grade, and pass/fail result.
- Implemented in the Exam class which extends Student.

c. Fee Module

- Accepts and displays tuition and exam fee details.
- Calculates total payable fee.
- Implemented in the Fee class which extends Exam.

d. Management Module

- Provides a menu-driven interface to:
 - Add and view student details
 - Add and view exam results
 - Add and view fee details
 - View full student profile (all details combined)
 - Exit the system

5. Features

- Modular class hierarchy using interface and inheritance.
- Real-time data entry and display via command line.
- Input validation (basic level).
- Clear separation of concerns using class responsibilities.
- Menu-driven user interaction.

6. Future Enhancements

- Add file handling or database support for persistent storage.
 - Introduce a GUI using JavaFX or Swing.
 - Add login-based access for admin or student users.
 - Improve input validation and error handling.
 - Export reports to PDF or Excel.
-

7. Conclusion

The *Student Management System* is a useful project to understand the application of OOP in solving real-life problems. It offers a structured way to manage academic data and can be expanded further for real-world use in schools and colleges

Code :-

```
import java.util.ArrayList;
import java.util.Scanner;

interface studentinterface {
    void student_input();
    void display();
}

class Student implements studentinterface {
    private int system_id;
    private String name;
    private String course;
    private int sem;

    @Override
    public void student_input() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Student Details");
        System.out.print("System ID: ");
```

```

        system_id = s.nextInt();
        s.nextLine(); // consume newline
        System.out.print("Name: ");
        name = s.nextLine();
        System.out.print("Course: ");
        course = s.nextLine();
        System.out.print("Semester: ");
        sem = s.nextInt();
    }

```

@Override

```

public void display() {
    System.out.println("\nStudent Details:");
    System.out.println("System ID: " + system_id);
    System.out.println("Name: " + name);
    System.out.println("Course: " + course);
    System.out.println("Semester: " + sem);
}

```

```

public int getSystemId() {
    return system_id;
}
}

```

```

class Exam extends Student {
    protected int sub1 , sub2, sub3, sub4, sub5;
    protected int total;
    protected float per;
    protected String grade;
    protected String result;

    public void examinput() {
        Scanner s = new Scanner(System.in);
    }
}

```

```

        System.out.println("\nEnter Marks:");

        System.out.print("Subject 1: "); sub1 = s.nextInt();
        System.out.print("Subject 2: "); sub2 = s.nextInt();
        System.out.print("Subject 3: "); sub3 = s.nextInt();
        System.out.print("Subject 4: "); sub4 = s.nextInt();
        System.out.print("Subject 5: "); sub5 = s.nextInt();

        total = sub1 + sub2 + sub3 + sub4 + sub5;
        per = total / 5.0f;

        if (per >= 90) grade = "A";
        else if (per >= 75) grade = "B";
        else if (per >= 60) grade = "C";
        else if (per >= 50) grade = "D";
        else grade = "F";

        result = (sub1 >= 35 && sub2 >= 35 && sub3 >= 35 && sub4 >= 35 && sub5 >= 35) ? "Pass"
: "Fail";
    }

    public void examDisplay() {
        if (sub1 == -1) {
            System.out.println("\nNo exam data available.");
            return;
        }
        super.display();
        System.out.println("\nExam Details:");
        System.out.println("Total Marks: " + total);
        System.out.println("Percentage: " + per + "%");
        System.out.println("Grade: " + grade);
        System.out.println("Result: " + result);
    }
}

```

```

class Fee extends Exam {
    private float tuitionFee = -1, examFee, totalFee;

    public void feeInput() {
        Scanner s = new Scanner(System.in);
        System.out.println("\nEnter Fee Details:");
        System.out.print("Tuition Fee: "); tuitionFee = s.nextFloat();
        System.out.print("Exam Fee: "); examFee = s.nextFloat();
        totalFee = tuitionFee + examFee;
    }

    public void feeDisplay() {
        if (tuitionFee == -1) {
            System.out.println("\nNo fee details available.");
            return;
        }
        System.out.println("\nFee Structure:");
        System.out.println("Tuition Fee: ₹" + tuitionFee);
        System.out.println("Exam Fee: ₹" + examFee);
        System.out.println("Total Fee: ₹" + totalFee);
    }

    @Override
    public void display() {
        super.display();
        examDisplay();
        feeDisplay();
    }
}

```

```

public class StudentManagementSystem1 {
    static ArrayList<Fee> students = new ArrayList<>();
}

```

```

static Scanner sc = new Scanner(System.in);

public static void main(String[] args) {
    int ch;
    do {
        System.out.println("\nStudent Management ");
        System.out.println("1. Add Student Details");
        System.out.println("2. View Students Details");
        System.out.println("3. Add Exam Marks Details");
        System.out.println("4. View Exam Result Details");
        System.out.println("5. Add Fee Details Details");
        System.out.println("6. View Fee StructureDetails");
        System.out.println("7. View All Details of student");
        System.out.println("8. Exit");
        System.out.print("Enter your choice: ");
        ch = sc.nextInt();

        switch (ch) {
            case 1:
                Fee f = new Fee();
                f.student_input();
                students.add(f);
                System.out.println("Student Added Successfully!");
                break;

            case 2:
                for (Fee stu : students) {
                    ((Student) stu).display();
                }
                break;

            case 3:
                System.out.print("Enter System ID to Add Marks: ");

```

```
int id1 = sc.nextInt();  
boolean marksFound = false;  
for (Fee stu : students) {  
    if (stu.getSystemId() == id1) {  
        stu.examinput();  
        System.out.println("Marks Entered Successfully!");  
        marksFound = true;  
        break;  
    }  
}  
if (!marksFound) System.out.println("Student Not Found!");  
break;
```

case 4:

```
System.out.print("Enter System ID to View Exam Result: ");  
int id2 = sc.nextInt();  
boolean foundExam = false;  
for (Fee stu : students) {  
    if (stu.getSystemId() == id2) {  
        stu.examDisplay();  
        foundExam = true;  
        break;  
    }  
}  
if (!foundExam) System.out.println("Student Not Found!");  
break;
```

case 5:

```
System.out.print("Enter System ID to Add Fee Details: ");  
int id3 = sc.nextInt();  
boolean feeAdded = false;  
for (Fee stu : students) {  
    if (stu.getSystemId() == id3) {
```



```
        stu.feeInput();  
        System.out.println("Fee Details Entered Successfully!");  
        feeAdded = true;  
        break;  
    }  
}  
if (!feeAdded) System.out.println("Student Not Found!");  
break;
```

case 6:

```
System.out.print("Enter System ID to View Fee Structure: ");  
int id4 = sc.nextInt();  
boolean foundFee = false;  
for (Fee stu : students) {  
    if (stu.getSystemId() == id4) {  
        stu.feeDisplay();  
        foundFee = true;  
        break;  
    }  
}  
if (!foundFee) System.out.println("Student Not Found!");  
break;
```

case 7:

```
System.out.print("Enter System ID to View All Details: ");  
int id6 = sc.nextInt();  
boolean foundAll = false;  
for (Fee stu : students) {  
    if (stu.getSystemId() == id6) {  
        stu.display();  
        foundAll = true;  
        break;  
    }  
}
```

```
    }  
    if (!foundAll) System.out.println("Student Not Found!");  
    break;  
  
    case 8:  
        System.out.println("Exiting System. Goodbye!");  
        break;  
  
    default:  
        System.out.println("Invalid Choice. Try again.");  
    }  
} while (ch != 8);  
}  
}
```

Output:-

Student Management

1. Add Student Details
2. View Students Details
3. Add Exam Marks Details
4. View Exam Result Details
5. Add Fee Details Details
6. View Fee StructureDetails
7. View All Details of student
8. Exit

Enter your choice: 1

Enter Student Details

System ID: 101

Name: Rahul

Course: BCA

Semester: 4

Student Added Successfully!

Student Management

Enter your choice: 3

Enter System ID to Add Marks: 101

Enter Marks:

Subject 1: 80

Subject 2: 75

Subject 3: 90

Subject 4: 85

Subject 5: 70

Marks Entered Successfully!

Student Management

Enter your choice: 5

Enter System ID to Add Fee Details: 101

Enter Fee Details:

Tuition Fee: 20000

Exam Fee: 1500

Fee Details Entered Successfully!

Student Management

Enter your choice: 7

Enter System ID to View All Details: 101

Student Details:

System ID: 101

Name: Rahul

Course: BCA

Semester: 4

Exam Details:

Total Marks: 400

Percentage: 80.0%

Grade: B

Result: Pass

Fee Structure:

Tuition Fee: ₹20000.0

Exam Fee: ₹1500.0

Total Fee: ₹21500.0

Student Management

Enter your choice: 8

Exiting System. Goodbye!

