SRM INSTITUTE OF SCIENCE AND TECHNOLOGY - RAMAPURAM

PROJECT TITLE :

STUDENT RESULT PROCESSING SYSTEM

Java Programming (21CSE269T)

DEPARTMENT : COMPUTER SCIENCE AND BUSINESS SYSTEMS

Submitted By:

ATHARVA SALUNKHE

RA2411042020005

OBJECTIVE

The main objective of this project is to build a simple Student Result Processing System using core Java and file handling. The purpose is to make managing student academic records easier, faster, and more accurate compared to manual methods.

Instead of calculating marks and grades by hand, this system automates the entire process. It calculates the total marks, average score, and final grade based on the marks entered for each subject. This reduces calculation mistakes and saves time.

The system also stores student information in a text file, so the data remains saved even after the program is closed.

Whenever needed, the stored records can be retrieved and displayed. Users can search for a specific student by entering the roll number, making it easy to find individual records quickly.

Another important feature of the system is identifying students who have failed in any subject or whose overall performance does not meet the required standard. This helps in quickly analyzing academic results.

Overall, this project shows how object-oriented programming principles and file handling concepts in Java can be applied to create a practical and useful application for managing student results efficiently.

PROBLEM DESCRIPTION

In many schools and colleges, student results are still prepared manually or maintained in spreadsheets. This method often leads to several practical problems.

Manual calculations of total marks and averages can easily result in mistakes. A small error in addition or percentage calculation can affect the final grade. Assigning grades manually also increases the risk of incorrect evaluation.

Searching for a specific student's record in registers or large spreadsheets is time-consuming. As the number of students increases, managing and organizing data becomes more difficult. Manual record keeping also increases the chances of data being misplaced, damaged, or permanently lost due to improper storage.

Identifying students who have failed in one or more subjects is another challenge. Teachers may need to review each record individually, which is inefficient and slow.

To overcome these issues, an automated system is required. Such a system should:

- Store student information in a structured manner.
- Automatically calculate total marks, average, and grade.
- Save records permanently using file handling.
- Provide quick search functionality based on roll number.
- Display the list of failed students easily.

The proposed Student Result Processing System is designed to address these problems. Built using core Java, it offers a simple, reliable, and user-friendly way to manage academic records efficiently while reducing errors and saving time.

CLASS DESIGN:

STUDENT CLASS

This is the main class that represents a single student and contains all academic details.

Attributes:

1. int rollNo – Unique roll number of the student.
2. String name – Name of the student.

3. int m1, m2, m3, m4, m5 – Marks of five subjects.

Methods:

1. **getTotal()** – Calculates total marks of five subjects.

2. **getAverage()** – Calculates average marks.

3. **getGrade**() – Assigns grade based on average marks.

4. **isFailed()** – Checks whether the student has failed (any subject < 50 or average < 50).

5. **display()** – Displays student details including total, average, and grade.

6. **display(boolean showMarks)** – Overloaded method to display detailed marks.

7. **toFileString()** – Converts student object into a comma-separated string format for storing in file.

MAIN CLASS

This class controls the entire program and provides a menu-driven interface for user interaction.

Attributes:

1. String FILE_NAME – Name of the file (students.txt) where records are stored.

Methods:

1. **addStudent()** – Takes student input and writes it to file.
2. **displayAll()** – Reads all student records from file and displays them.
3. **searchStudent(int rollNo)** – Searches for a student using roll number.
4. **displayFailed()** – Displays students who failed.
5. **main() –** Contains menu-driven logic using switch-case and loop.

APPROACH USED

The system is designed using a file-based sequential processing method.

Each student's information is saved in a text file, where one line represents one complete record. The data is stored in a comma-separated format so that it remains structured and easy to read.

For example:
101,Arun,78,65,90,88,56

Here, the values represent roll number, name, and subject marks in a fixed order.

Whenever the program needs data, it performs the following steps:

- Opens the file and reads it line by line.
- Splits each line using a comma as the separator.
- Converts the extracted values into appropriate data types.
- Recreates the Student object using these values.
- Calculates total marks, average, and grade.
- Displays the required output to the user.

This approach offers several advantages:

- It is simple to implement using only core Java concepts.
- It does not require any database setup or external tools.
- Data remains stored permanently through file handling.
- The logic is clear and easy to understand, making it suitable for academic learning and beginner-level projects.

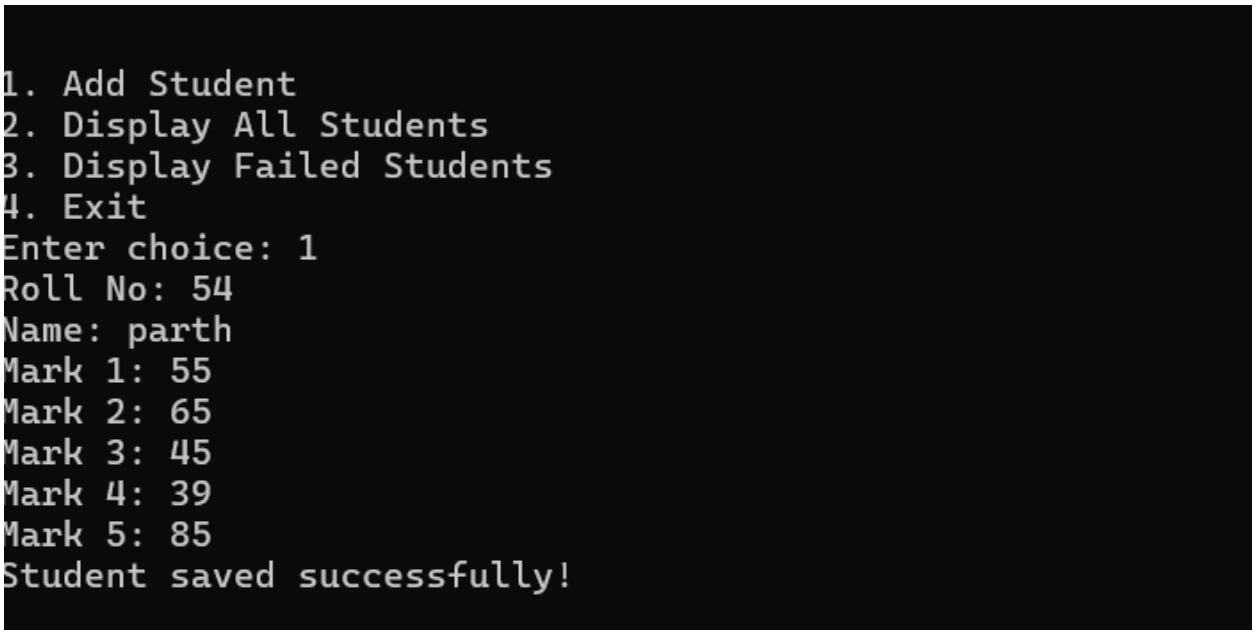
Overall, this method keeps the system lightweight, structured, and practical for small-scale result management.

OUTPUT:

TEST CASE 1: above 40 marks

```
C:\Users\salun\Downloads\StudentResultSystem>java StudentResultSystem

1. Add Student
2. Display All Students
3. Display Failed Students
4. Exit
Enter choice: 1
Roll No: 9
Name: yu
Mark 1: 54
Mark 2: 45
Mark 3: 56
Mark 4: 45
Mark 5: 55
Student saved successfully!

1. Add Student
2. Display All Students
3. Display Failed Students
4. Exit
Enter choice: 3

Roll No: 99
Name: test
Total: 330
Average: 66.0
Grade: F
--------------------------

1. Add Student
2. Display All Students
3. Display Failed Students
4. Exit
Enter choice:
```

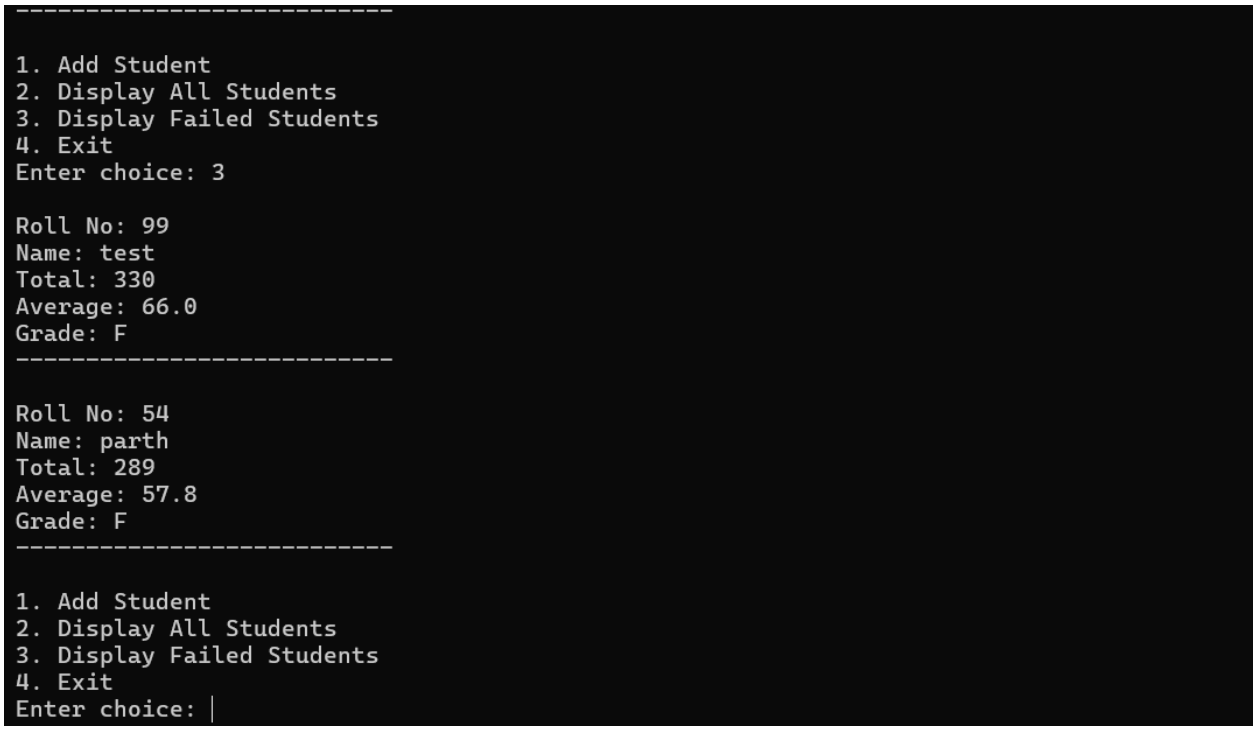
Test case 2 : below 40 in one subject



Test case 3 : display all students

```
1. Add Student
2. Display All Students
3. Display Failed Students
4. Exit
Enter choice: 2

Roll No: 45
Name: 55
Total: 294
Average: 58.8
Grade: D
--------------------------

Roll No: 99
Name: test
Total: 330
Average: 66.0
Grade: F
--------------------------

Roll No: 9
Name: yu
Total: 255
Average: 51.0
Grade: D
--------------------------

Roll No: 54
Name: parth
Total: 289
Average: 57.8
Grade: F
--------------------------

1. Add Student
2. Display All Students
3. Display Failed Students
4. Exit
Enter choice:
```

Test case 4: display failed student



CONCLUSION

Git hub link :
https://github.com/Atharvasalunkhe881/Student-Result-Processing-System

The Student Result Processing System clearly shows how core Java concepts can be applied to solve a real-world problem. In this project, important features of Java such as classes and objects are used to represent student data in

a structured way. Constructors help in initializing student details, while methods are used to perform calculations and other operations. Concepts like file handling ensure that records are stored permanently, and exception handling improves the reliability of the program by managing possible runtime errors. A menu-driven structure makes the system interactive and easy to operate.

By automating the calculation of total marks, average, and grade, the system reduces manual effort and minimizes errors. It also makes it easier to search for specific students and quickly identify those who have failed in any subject.

Overall, this project strengthens practical understanding of object-oriented programming principles and demonstrates how file handling can be used effectively to build a simple but useful Java application.