

**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 primary goal of this project is to develop a simple file-based Student Result Processing System using core Java concepts. The system is designed to manage student academic records efficiently.

Specifically, the system aims to:

Automate Student Record Management: Replace manual calculation of marks and grades with an automated program to reduce errors.

Calculate Academic Performance: Compute total marks, average, and grade automatically based on subject marks.

Store and Retrieve Data: Save student records in a text file and retrieve them whenever required.

Search Functionality: Allow users to search student details using roll number.

Identify Failed Students: Display students who fail in any subject or overall performance.

Ensure Data Accuracy: Maintain correct and structured student records using file handling techniques.

The project demonstrates the practical implementation of object-oriented programming and file handling concepts in Java.

PROBLEM DESCRIPTION

In many educational institutions, student results are calculated manually or maintained using spreadsheets. This approach creates multiple challenges such as:

Calculation errors in total and average marks.

Incorrect grade assignment.

Difficulty in searching student records quickly.

Manual record keeping that is time-consuming.

Risk of data loss due to improper storage.

Difficulty in identifying failed students efficiently.

When the number of students increases, manual result processing becomes inefficient and error-prone.

Searching for a particular student record requires scanning multiple files or sheets manually.

There is a need for an automated system that can:

Store student details properly.

Calculate total, average, and grade automatically.

Save records permanently using file handling.

Allow easy searching of student data.

Display failed students quickly.

The proposed Student Result Processing System provides a simple, reliable, and user-friendly solution to manage student academic records using core Java programming.

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 follows a File-Based Sequential Processing Approach.

Each student record is stored in a text file as a single line using comma-separated values.

Example record format:

101,Arun,78,65,90,88,56

Whenever data is required:

The program reads the file line by line.

Splits the string using comma.

Reconstructs the Student object.

Performs calculations and displays results.

This approach ensures:

Simple implementation using core Java.

No dependency on database.

Permanent storage using file handling.

Easy understanding for academic learning purposes.

CONCLUSION

The Student Result Processing System successfully demonstrates the implementation of core Java concepts such as classes, objects, constructors, method overloading, file handling, exception handling, and menu-driven programming.

The system reduces manual effort in managing student results and ensures accurate calculation of total, average, and grade. It also allows efficient searching and identification of failed students.

This project enhances understanding of object-oriented programming and practical file handling in Java.