

GALGOTIAS UNIVERSITY

CGPA CALCULATOR

Java-Based Academic Performance System

Team: Eternal Coders

Team Leader: Tausif Hassan

Members: Anjali Kumari, Dipra Khurana

01 / Introduction

What is the CGPA Calculator?

It is a dedicated desktop application built in Java, designed to provide students with an immediate, accurate assessment of their academic standing by calculating the Cumulative Grade Point Average (CGPA).



Java Power

Platform independence and robustness for secure arithmetic operations.

Technology Stack

- ✓ **Language:** Core Java SE (JDK 17+)
- ✓ **GUI Framework:** Java Swing (or similar library)
- ✓ **Paradigm:** Object-Oriented Programming (OOP)



Desktop Application

Provides a reliable, quick-access utility without needing web connectivity.

02 / Problem & Solution

The Challenge

- ✓ **Inaccuracy Risk:** Manual spreadsheet or paper calculations are highly susceptible to errors.
- ✓ **Non-Standardized:** Grade conversion and credit weighting can vary, leading to confusion.
- ✓ **Lack of Real-Time Feedback:** Students cannot easily model "what-if" scenarios for future performance.

Our Solution (Key Features)

- ✓ **Weighted Average Engine:** Handles variable credit hours accurately.
- ✓ **Grade Mapping:** Built-in logic to convert letter grades to numerical grade points automatically.
- ✓ **User Interface (GUI):** Simple forms allow quick entry of course data (Course, Credits, Grade).

03 / Architecture & OOP Concepts

Modular Class Design

The application is structured around dedicated classes to enforce Single Responsibility Principle (SRP):

- ✓ **Course:** Encapsulates course data (Name, Credit, Grade).
- ✓ **Semester:** Manages a collection of Course objects.
- ✓ **CalculatorEngine:** Contains the core business logic for mathematical processing.



OOP Pillars

ENCAPSULATION

Data protection via private fields.

ABSTRACTION

Hiding complex calculation logic.

04 / Mathematical Foundation

Weighted Average Formula

The CGPA is computed as the total weighted grade points divided by the total credit hours earned across all included semesters.

$$\text{CGPA} = \frac{\sum_{i=1}^n C_i \times P_i}{\sum_{i=1}^n C_i}$$

\$C_i\$: Credit hours for course \$i\$ | \$P_i\$: Grade Point for course \$i\$

05 / Implementation Snippet

The Calculation Loop

The engine iterates over all courses, multiplying credits by points, and handles edge cases like division by zero.

```
// Core Logic within CalculatorEngine.java

public double calculateCGPA(List courses) {

    double totalPoints = 0.0;

    int totalCredits = 0;

    for (Course course : courses) {

        double gradePoint = GradeMapper.getPoint(course.getGrade());
```



06 / Future Scope & Expansion

Phase 2: Data Persistence



Local Storage / Serialization

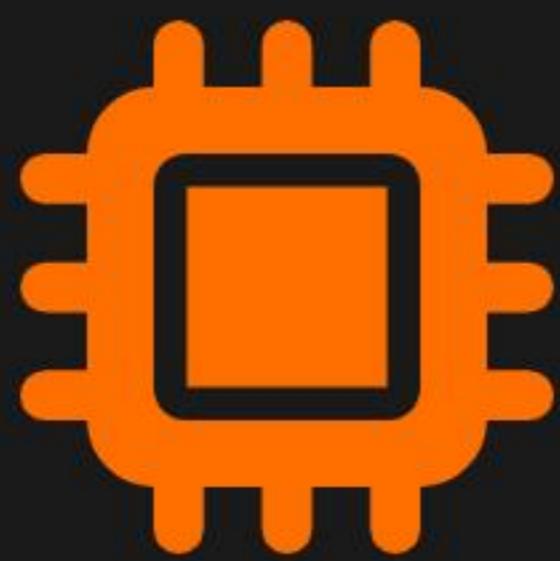
Implement Java Serialization or simple file I/O to save student data locally, allowing history retrieval.

Phase 3: Extended Features



Graphical Reporting

Integrate charting libraries to visualize semester-to-semester performance trends.



THANK YOU

QUESTIONS?

Team Eternal Coders - Galgotias University