# CH University - ODLSTUDENTS GRADE GENERATOR

FOR CALCULATION STUDENTS GRADES BASES ON 3 SUBJECTS

## A PROJECT REPORT

Submitted by:

**Anurag (22BCA10058)**

Submitted to:

**MR Rishabh.**

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR IN COMPUTER APPLICATIONS**

Chandigarh University

January – May 2025

**Project Title:**

Student Grade Management System **Objective:**

The main objective of this mini project is to develop a desktop-based Java application that facilitates the management of student academic records in an efficient and user-friendly manner.

It allows entry of student data, calculation of total and average marks, and determines pass or fail status based on input scores.

This project also provides an opportunity to explore Java Swing for GUI design, understand Object-Oriented Programming (OOP) principles, handle user events, and apply exception handling for robust applications.

**Tools and Technologies Used:**

* Programming Language: Java (JDK 8 or higher)
* Development Environment: IntelliJ IDEA (You may also use Eclipse, NetBeans)
* GUI Framework: Java Swing (AWT components also possible)
* File System: Local file storage (optional)
* Concepts Used:
* Classes and Objects (OOP)
* Arrays and Loops
* GUI Programming using Swing components (JFrame, JLabel, JTextField, JButton, JTextArea)
* Exception Handling using try-catch - Event Handling with ActionListener

**System Requirements:**

* Operating System: Windows/Linux/Mac OS
* Java Development Kit (JDK): Version 8 or above
* RAM: Minimum 2GB (Recommended 4GB+)
* Disk Space: Minimum 100MB for IntelliJ IDEA and Java project files**Project Modules:**

1. Student Input Form:

* Collects essential data: Student Name, ID, Marks for 3 Subjects.
* Ensures clean and organized layout for user interaction.

2. Grade Calculation Module:

* Sums up the marks to get a total score.
* Calculates the average score from three subjects.
* Evaluates whether the student passed or failed based on minimum marks.

3. Output Display:

* Results are printed in a scrollable text area for better readability.
* The output includes Name, ID, Total Marks, Average, and Result Status.

4. Error Handling:

* The application handles non-integer inputs using try-catch.
* Ensures that the application does not crash on invalid data.

5. Future Enhancements (Optional):

* Integration with file I/O to save/load student records.
* Generate downloadable report cards.
* Data visualization for class performance trends.

**Sample Output:**

Example Output for Two Students:

John (S101) - Total: 230, Average: 76.67, Result: Pass

Jane (S102) - Total: 165, Average: 55.00, Result: Fail

This output demonstrates how the application processes inputs, performs calculations, and presents results clearly.

**Code Overview:**

Main Class: StudentGradeManager.java

This is the primary class that builds the Swing interface using JFrame, JLabel, JTextField, JButton, and JTextArea. It includes event handling to process user actions such as clicking the "Add Student" button.

Inner Class: Student.java

This class models a student entity with attributes: name, ID, and marks. It includes methods to calculate total, average, and determine pass/fail.

Key Java Concepts Used:

* JFrame: Base window for GUI
* JLabel: Text labels for fields
* JTextField: Input fields for user data
* JButton: Interactive button to trigger actions
* JTextArea: Output display area
* JScrollPane: Scrollable container for JTextArea
* ActionListener: Captures and handles button clicks
* try-catch: Handles exceptions during input parsing

**Conclusion:**

This project serves as a fundamental learning experience for students who are beginning to explore Java programming and desktop application development.

Key Learning Outcomes:

* Building interactive GUIs using Java Swing
* Writing modular and object-oriented code
* Implementing input validation and exception handling
* Displaying formatted output based on calculations

Overall, this project lays the groundwork for more complex applications in the future and can be extended further by integrating databases, report generation, or web-based components.