

# **B202 Advance programming Report On Grade tracking application**

**Student id – GH1028457 , Harsimran Singh**

## **Grade tracking Application**

- 1. Introduction :-** Java is most used language which is used by users to built application. The purpose of this project is to make student grade tracking application in which they can track their records. This project was made by using Java programming language. The application is console based application that manages student record such as storing grades , calculate averages etc. The code runs in command line interface.
- 2. Objectives of the Project :-** The main Objectives of this project are
  - To build a working java console application
  - To use object oriented programming concepts
  - To easily store and manage student records
  - To allow users to edit their records such as add or delete
  - To calculate average of students
  - Using file handling we can store and restore student data
  - To create a system that is easy to maintain
- 3. Technologies Used in the project:-** As per module guidelines , Java is used in this project.
  - Java
  - Object Oriented Programming(OOP) concepts
  - Java file handling (java.io )
- 4. Application Description :-** The application is basically grade tracking application which is console based management system. Users can interact with the system through a menu which is displayed in the console.

The system allows the user to

  - Add new student
  - Add grades

- Show all student records
- Save student records
- Edit student's name
- Delete student
- Delete all student record
- Save records

**5. Application design:** - The application design is object oriented and it is divided into three main sections.

### **5.1 Student Class**

The student class shows an individual student which stores

- Student Id
- Student name
- Grades

It also contains methods which

- Add grades
- Calculate average
- Shows pass or fail status
- Shows student information

### **5.2 GradeManager Class**

This class manages all operation which is related to students .

- Adding Students
- Editing students
- Delete Students
- Clear data
- Save data
- Loading data from file

This class is core logic of the application.

### **5.3 Application class**

The application class contains the main() method which

- Display menu
- Take input from user
- Call methods from GradeManager

This class is main class which controls flow of the program

## **6. Meaning and purposes of classes:-**

### **6.1 Student Class:-** The student class shows each student in the system.

It stores all data which is related to student. The student class also contains variables which stores student id , student name as well as grades list. The grades are stored in ArrayList which helps to add multiple grades at once without any fix size.

When a new student is built , the constructor of the class is used to set student id and name. At the same time, the grades list is set to ensure it is ready to store values. Getter method are used to access the ID , name as well as grades of the student. Setter is used to allow editing student name which ensures regulated access to student data.

A method for adding grades for a student allow new grades to be stored . Another method calculates the averages by doing average which is sum of all grades dividing by total number of grades. If there were no grades , the method returns zero to avoid errors in application. The class also include logic to decide whether student has passed or fail based on calculated average. In last , The toString() method is overridden to show student details in efficient way when it is displayed in console.

### **6.2 GradeManager Class :** - Every process pertaining to students are managed by the GradeManager class. It acts as the application's main operational logic layer.

This class allows the at once handling of several students by collecting all student objects in an ArrayList. There are ways to add new students while making sure that duplicate student IDs are prevented. Students are found using an internal search strategy based on their ID.

Additionally, the class offers the capacity to add grades, modify student names, remove specific students, and remove all student data. The user may fully handle the student data thanks to these

procedures.

One of this class's major duties is handling files. In order to prevent information from being lost upon program end, the application saves student data to a text file. The ID, name, and grades of every student are kept in an organized format.

The file is accessed when the application launches, and the stored data is used to create new student objects. It makes it possible for the system to effortlessly recover earlier records. The application is kept from crashing due to file-related errors through basic exception handling.

**6.3 Application Class :-** The Application class contains the main() method, which acts as the program's starting point. This class is in control of user interaction and flow of the program.

When the application launches, current information is loaded to the file and a GradeManager class object is created. The application shows a menu where user can choose different type of functions and also until user chooses to stop or exit , a menu will show every time. There is framework called switch case framework which manages various menu which include add students , adding grade as well as editing data. The scanner class is used to read input from users.

## **7. Menu System Overview:-**

The menu system lets users to communicate with the program in a clear and simple way.

Each menu option relates to a particular action, such as adding a student or adding grades, viewing all records, editing names, removing students, or deleting all data. The switch-case structure enhances readability while making future program extensions easier. More menu options enable the addition of new features without significantly changing the current logic.

**8. File Handling Explanation:** File handling is implemented by using standard input as well as output classes of java. File handling helps to maintain structure so well so that it can easily read later. when terminal

runs firstly it checks the file existence , if the file is found the data will be loaded. This make sure that system keeps the data across program executions . This method helps to avoid use of external databases while providing efficient storage

## **9. Implementation details**

The application is designed using Java features

- Array list is used to store student data
- Scanner is used to take input from the console
- BufferedReader and FileWriter are used for file handling
- Loops and conditional statements are used for menu directions

The program also ensure

- Preventing duplicate student id
- Check valid grade ranges which is 100
- Handle invalid menu input

## **10.File Handling**

To maintain data efficiently , the application uses text file which is student.txt

When program starts , it takes existing data from file. When user save or delete or exits , the data is written back to the file.

## **11.Sample Output of application**

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Users\singh\AppData\Local\Temp\jvarkit\jvarkit.jar" -Djava.class.path=C:\Users\singh\AppData\Local\Temp\jvarkit\jvarkit.jar  
Loaded successfully.
```

```
===== Welcome!! Student Grade Tracking Application =====
```

1. Add Student which you have to add into the data
2. Add Grade
3. Show All Students who are added
4. Save student record
5. Delete student from record
6. Edit Student name
7. Delete all data
0. Exit

```
Choose:
```

## 12. Challenges Faced while making Application

Some challenges faced while developing are

- Managing user input correctly using scanner
- Handling input and output without errors
- Avoiding duplicate student
- Program should not crash due to invalid input

These all challenges were solved using input validation, conditional checks and structured programming

## 13. Conclusion

To conclude , Grade tracking application meets all the requirements according to assessment brief. It also shows use of java programming and object oriented fundamentals.

Classes, objects, methods, collections, conditional statements as well as loops were applied effectively. This design make it easier to extend application by adding new features. The use of file handling allows application to store data between program executions which makes it more real as compare to temporary in memory system.

This project also helped to improve my problem skills in handling user input as well as managing program flow in a console terminal. Overall the project meets all the requirements and also serves example of how

java can be used to build simple management system. The knowledge and experience developed by this project gives idea to make more complex java application in future.

#### **14. References:-**

*JDK 25 Documentation* (no date) *Oracle Help Center*. Available at: <https://docs.oracle.com/en/java/javase/25/> (Accessed: December 16, 2025).

*The Java™ Tutorials* (no date). Available at: <https://docs.oracle.com/javase/tutorial/> (Accessed: December 16, 2025).

**15. Github Repository :-** The code of project is uploaded in GitHub repository which can be found under src/gradeapp and also I uploaded video demonstration link for my code under readme file.

Repository :- [singhunique/GradeTrackerApplication: This is grade tracker application which tracks grades of student](#)