C++ DEVELOPMENT INTERNSHIP (TASK 1)

Project Title: Student Grade Calculator

Project Description: Build a C++ program that allows users to input student information and their respective scores for assignments, quizzes, and exams. The program will calculate the overall grade based on user-defined weightage for each category and display the final grade along with any additional information like letter grades or comments.

Steps you can Implement:

User Interface: Create a user-friendly command-line interface that allows users to input student details (name, ID, etc.) and their scores for assignments, quizzes, and exams.

Data Validation: Implement data validation to ensure that user inputs are within valid ranges (e.g., scores between 0 and 100) and handle erroneous inputs gracefully.

Grade Calculation: Design the algorithm to calculate the overall grade based on userdefined weightage for assignments, quizzes, and exams. You can use a weighted average calculation.

Letter Grade Mapping: Map the calculated numerical grade to a letter grade (A, B, C, etc.) according to a grading scale.

Comments or Feedback: Provide additional information, such as comments or feedback, based on the final grade. For example, you can offer congratulations for high grades or suggest improvements for lower grades.

Loop for Multiple Students: Allow the program to handle multiple students' information and grades by implementing a loop. After processing one student, ask if the user wants to input data for another student.

Display Results: Display the final grade, letter grade, and any comments for each student.

File Input/Output (Optional): Implement the ability to save student data and grades to a file and load data from a file for future reference.

Modularization: Organize your code into functions to make it more maintainable and readable. For example, you can have separate functions for input, calculation, and output.

Error Handling: Handle exceptions and errors gracefully. Ensure that the program doesn't crash due to unexpected inputs or issues.

Testing: Test your program with various scenarios, including different weightage configurations, numbers of students, and input values, to ensure its reliability.

Documentation: Document your code with comments and provide a user guide that explains how to use the program.