**Technical Design Document**

**Name:** Khoa Duong  
**Date Created:** March 8, 2025

**Program Description:**

This program allows a teacher to input students' names and exam grades, store the data in a CSV file (grades.csv), and display the stored data in a formatted table (The program also calculates and records the average of the three exam scores).

**Functions used in the Program (list in order as they are called):**

**1. Function Name: write\_grades()**

* **Description:**  
  This function collects student data (first name, last name, and three exam scores), calculates the average score, and writes the data into a CSV file (grades.csv).
* **Parameters:**
  + None (data is collected via user input).
* **Variables:**
  + num\_student (int): Stores the number of students to be entered.
  + first\_name (str): Stores the student's first name.
  + last\_name (str): Stores the student's last name.
  + Exam\_1, Exam\_2, Exam\_3 (int): Stores the grades for three exams.
  + Average : Stores the calculated average of the three exams.
* **Logical Steps:**
  + Prompt the user to enter the number of students.
  + Open grades.csv for writing and write the header row.
  + Loop through the number of students:
    - Collect their first name, last name, and three exam scores.
    - Calculate the average score:
  + Write the collected student data into the CSV file.
  + Close the file and confirm that data has been saved.
* **Returns:**
  + None (writes data to grades.csv).

**2. Function Name: main()**

* **Description:**  
  This function executes the main program workflow by calling write\_grades() to collect data and read\_grades() to display it.
* **Parameters:**
  + None.
* **Variables:**
  + reader (csv.reader object): Reads the CSV file.
  + data (list): Stores the content of the CSV file as a list of rows.
* **Logical Steps:**
  + Call write\_grades() to collect and save student data.
  + Open grades.csv for reading.
  + Convert the file contents into a list.
  + Print the header row formatted in columns.
  + Print a separator line.
  + Loop through the student records and print each row formatted in columns.
* **Returns:**
  + None (prints student data to the console).

**Logical Steps:**

1. **Call write\_grades()** → Collects student information and writes it to grades.csv.
2. **Read grades.csv** → Converts data into a list.
3. **Print formatted table** → Displays student records in tabular format.

**Link to Repository:** <https://github.com/khoakhi3/COP2373>

A screenshot of a computer program

AI-generated content may be incorrect.