**Technical Design Document: Programming Exercise CSV**

**Name:** Carson Harbin  
**Date Created:** March 23rd, 2025

**Program Description:**

This program allows an instructor to enter student grades into a CSV file named grades.csv. The instructor inputs the number of students, along with each student’s first name, last name, and three exam scores. The program then writes this information into the CSV file with a structured header. Additionally, another function reads the file and displays the data in a tabular format. The program utilizes Python’s built-in csv module and ensures proper file handling using the with statement.

**Functions Used in the Program (Listed in Order of Execution):**

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

* **Description:** This function collects student information and writes it to a CSV file.
* **Parameters:** None.
* **Variables:**
  + filename (string): The name of the file where grades are stored (grades.csv).
  + num\_students (integer): The number of students to be recorded.
  + first\_name, last\_name (string): The student’s first and last name.
  + exam1, exam2, exam3 (integers): The student’s three exam scores.
* **Logical Steps:**
  1. Open grades.csv in write mode.
  2. Write a header row: First Name, Last Name, Exam 1, Exam 2, Exam 3.
  3. Prompt the user for the number of students.
  4. Loop through the number of students and collect their details.
  5. Write each student’s data as a row in the CSV file.
  6. Close the file automatically using the with statement.
* **Returns:** None.

**2. Function Name: read\_grades()**

* **Description:** This function reads the grades.csv file and displays its contents in a formatted tabular structure.
* **Parameters:** None.
* **Variables:**
  + filename (string): The name of the file where grades are stored (grades.csv).
  + header (list): The first row containing column titles.
  + row (list): A single student's data extracted from the file.
* **Logical Steps:**
  1. Open grades.csv in read mode.
  2. Read the header row and print it formatted.
  3. Iterate over each row and print the student’s information in tabular format.
  4. Close the file automatically using the with statement.
* **Returns:** None.

**Logical Steps of the Program:**

1. Prompt the instructor for the number of students.
2. Collect and store student names and grades.
3. Write the data into grades.csv.
4. Read and display the stored data in tabular format.

**Testing Plan:**

|  |  |  |
| --- | --- | --- |
| INPUT TYPE | VALID EXAMPLE | INVALID EXAMPLE |
| Basic Student Entry | John Doe, 85, 90, 95 | John, 85, 90 (Missing last name and exam) |
| Multiple Students | Jane Smith, 78, 88, 92 and Tom Brown, 91, 87, 85 | , , , (Empty input) |
| Numeric Validity | Alex Green, 100, 95, 90 | Mark Blue, ninety, 85, 80 (Non-numeric grade) |
| File Readability | Display formatted student records correctly | Corrupt or missing file scenario |

**Link to repository:** https://github.com/CarsonHarbin/COP2373