**Technical Design Document**

**Name:** Matthew Pocrnic  
**Date Created:** 6/28/2025

### **Program Description:**

This program allows an instructor to enter data for a specified number of students, including their first name, last name, and three exam grades. The data is saved to a file named grades.csv. The program then reads this file and displays the contents in a table.

### **Functions used in the Program:**

#### **1. Function Name:** create\_grades\_file

**Description:** Collects student data from user input and writes it to a CSV file named grades.csv.

**Parameters:** None

**Variables:**

* num\_students (int): The number of students for whom data will be collected.
* first (str): The first name of the student.
* last (str): The last name of the student.
* exam1, exam2, exam3 (int): The grades for the three exams.
* writer (csv.writer): A writer object used to write rows to the CSV file.

**Logical Steps:**

1. Prompt the user to enter the number of students.
2. Open grades.csv in write mode using a context manager.
3. Create a CSV writer object.
4. Write the header row to the file.
5. For each student:  
   * Prompt for their first and last names.
   * Prompt for three exam grades, converting each to an integer.
   * Write a row to the CSV file with the collected information.

**Returns:** None

#### **2. Function Name:** display\_grades\_file

**Description:** Reads and displays the contents of grades.csv in a tabular format.

**Parameters:** None

**Variables:**

* reader (csv.reader): A reader object used to read data from the CSV file.
* row (list): A list representing each row of student data from the file.

**Logical Steps:**

1. Print table headers for the grade display.
2. Open grades.csv in read mode using a context manager.
3. Create a CSV reader object.
4. Skip the header row using next().
5. For each row in the file:  
   * Print the student data with formatted spacing for readability.

**Returns:** None

### **Logical Steps (Program Flow):**

1. The program starts by calling create\_grades\_file().
2. The user is prompted to input how many students' data they want to enter.
3. For each student, the program collects the name and exam grades and writes them to grades.csv.
4. After data entry is complete, display\_grades\_file() is called.
5. The program reads the CSV file, skips the header, and prints each student's data in a formatted table.
6. The program terminates after displaying the data.

**Link to my COP2373 repository:** [**here**](https://github.com/mpocrnic/COP2373)

Screenshot of output from running code



