**Instructor Grades Table**

**Name:** Austin Wilson

**Date Created:** 3/23/25

**Program Description:**

Uses two programs, one to create a csv file that receives and writes information for amount of students, each students name, and their grades on 3 exams. The next program reads the csv file and displays the information formatted as a table.

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

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

**Description:** Prompts instructor to input student details, (name, exam scores), and writes the data to a CSV file (grades.csv)

**Variables:** student\_total (int): Input received for total number of students instructor wishes to include on table

full\_name (str): The full name of the student inputted

name parts (list): list coming from the split full name

first name (str): first name extracted from name\_parts

last name (str): last name extracted from name\_parts

exam1, exam2, exam3 (int): number grades for a student on all three exams

file (file object): The opened CSV file where the data is written

writer (csv.writer): Object used to write rows to file

**Logical Steps:**

1. Asks instructor how many students to input data for

2. Opes grades.csv in write mode

3. Creates writer object and creates header row

4. Loops through number of students and receives their full name and splits it, then prompts for exam scores, then writes it to file

5. Closes csv file using with block

**Returns:** none, only performs writing actions

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

**Description:** reads the grades.csv file, processes, and displays the data as a table

**Variables:** file: Opened CS file

reader (csv.reader); CSV reader object used to read rows from the file

header (list): List containing the header row from the CSV file.

Row (list): A list representing a single row of student data read from the CSV file.

**Logical Steps:**

1. Opens grades.csv in read mode

2. Creats a CSV reader object

3. Skip header row calling next(reader)

4. Print header row in a table style

5. Loop through each row of data and print as table

6. Close reader.

**Returns: none,** only prints data

**Logical Steps:**

Run write\_grades function, and input data for student, which is then stored in the grades.csv file.

After running write\_grades, run read\_grades to read the student data stored in the previous program from the grades.csv file, displays it in a table format.

**Link to your repository:** www.yourrepository.com