**Name:** Jadyn Brabham  
**Date Created:** 03/18/2025

**Program Description:**This program is designed for an instructor to input grades of students for 3 exams. The program will store the data in a CSV file named grades.csv and allow the instructor to view the data in tabular format. The user will be prompted to input the number of students, their first name, last name, and grades for the 3 exams. Then it saves the data to the CSV file and displays the contents in a table.

**Functions used in the Program (list in order as they are called):**1. **Function Name:** main()  
**Description:** The main function calls both the write\_grades() and read\_grades() functions. It is the entry point of the program.

**Parameters:** None  
**Variables:** The main function does not have any variables, but the variables of the write\_grades() and read\_grades() functions are called within the main() function.

**Logical Steps:**

1. Call the write\_grades() function to input and store the data in the CSV file.
2. Call the read\_grades() function to display the content in a table.

**Returns:** None. The function initiates the flow of the program by calling the other functions.

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

**Description:** The write\_grades() function prompts the user for the number of students, their first and last name, and their grades on the 3 exams. If the user inputs invalid integers for the grades it displays a message and then exits the program. Then it writes the input data to the CSV file.   
**Parameters:** None

**Variables:**

1. num\_students: the number of students to be entered by the instructor
2. writer: a CSV writer object used to write data in the file
3. first\_name: the student’s first name input by the instructor
4. last\_name: the student’s last name input by the instructor
5. exam\_1: the grade for the first exam input by the instructor
6. exam\_2: the grade for the second exam input by the instructor
7. exam\_3: the grade for the third exam input by the instructor

**Logical Steps**:

1. Prompt the user to enter the number of students.
2. Open the grades.csv file in write mode.
3. Loop through the number of students and collect each student’s name and grades for the 3 exams.
4. Handle any invalid input in the exam grades and display an error message.
5. Write the student data into the CSV file.
6. Display a message to tell the user that the data has been written to the file.

**Returns:** None. The function does not return any values, but it writes the data to the CSV file.

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

**Description:** The read\_grades() function opens the grades.csv file in read mode, skips the header row, and displays the data in a tabular format. It skips the header row so that the header row is not displayed twice.

**Parameters:** None

**Variables:**

1. reader: a CSV reader object used to read data from the file
2. row: list containing the data for each student

**Logical Steps:**

1. Open the CSV file for reading.
2. Skip the header row to avoid displaying the column names as part of the table.
3. Print the header row with column names.
4. Iterate over each row of the CSV file, print the student’s data in a table.
5. Format the columns to ensure they are properly aligned.

**Returns:** None. The function does not return anything, but it does print the data from the file in a table.

**Logical Steps:**

1. Import the csv module
2. Call the main function()
3. Inside the main function(), the write\_grades() function is called.
4. After the grades are written to the CSV file, the main() function calls the read\_grades() function to display the data.

**Link to your repository:** <https://github.com/jbrabham23/COP2373/tree/master>

**Program Working**

**A screenshot of a computer

AI-generated content may be incorrect.**