**Name:** Estiward Casado Antigua

**Date Created:** March 22, 2025

**Program Description:**

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

1. **Function Name:** student\_info\_input()

**Description:** User inputs students’ information to store inside of grades.csv.

**Parameters:**

1. none

**Variables:**

* 1. rows\_list – list contains other lists that will be printed as each row to the grades.csv file
  2. row\_num (int) – input from user and is how many students that the user will input
  3. first\_name – input from user student’s first name
  4. last\_name – input from user student’s last name
  5. exam\_1\_grade – input from user student’s exam 1 grade
  6. exam\_2\_grade – input from user student’s exam 2 grade
  7. exam\_3\_grade – input from user student’s exam 3 grade
  8. row (list) – list that takes the 5 input and compiles into a list. The row list is then appended to the rows\_list.
  9. file – contains the grades.csv file opened in append mode
  10. writer – csv.writer class instance writing to file

**Logical Steps:**

1. Initialize rows\_list
2. While loop for input validation of row\_num.
   1. Row\_num is input, and then a try except block is run to convert it to an integer.
   2. If the input is valid, the loop breaks but if it isn’t, the user is prompted until the input is valid
3. For loop that runs as many times as the row\_num.
   1. The student information is input and then stored into the row list.
   2. For the exam grades while loops are run with try except blocks to ensure they are integers. If the input is valid the loop breaks and the program continues but if it isn’t the user is reprompted.
   3. The row list is then appended to the rows\_list.
4. The grades.csv file is opened in append mode.
5. A csv.writer class instance is created.
6. The csv.writer class’s writerows() method is used and writes the entirety of the rows\_list to the grades.csv file.

**Returns:**

1. none
2. **Function Name:** student\_info\_read()

**Description:** Open grades.csv file and display as table

**Parameters:**

1. none

**Variables:**

* 1. file – grades.csv file opened in read mode
  2. csv\_reader – instance of the csv.reader class
  3. line\_count (int) – used to determine what line the program is currently on to print the header only once

**Logical Steps:**

1. Open the grades.csv file in read mode.
2. Create an instance of the csv.reader class.
3. Initialize the line\_count variable as 0
4. For loop for each row in the grades.csv file
   1. To print the header, when the line\_count is 0, print the column names and dashes underneath them
   2. Add to the line count to stop printing the header again
   3. For the rest of the rows print each column
   4. Add to line count to ensure the variable is accurate in case it is needed in the future.

**Returns:**

1. none
2. **Function Name:** main()

**Description:** User is prompted for whether they want to see the data from grades.csv or add more data. Then, the respective function is called.

**Parameters:**

1. none

**Variables:**

* 1. choice (str) – Input from user that is used to choose which function to call

**Logical Steps:**

1. In a while loop for input validation.
   * 1. Prompt user to enter 1 or 2 to input student data or read it.
     2. If choice is not valid, an error message is printed and the loop continues.
     3. Once a valid answer is given, the loop breaks.
2. If the choice is ‘1’, the student\_info\_input() function is run.
3. If the choice is ‘2’, the student\_info\_read() function is run.

**Returns:**

1. none

**Logical Steps:**

1. Import csv module
2. Call main(). Main() calls student\_info\_read() or student\_info\_read().
3. Program stops.

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

A screen shot of a computer

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.