**CSV Exercise Technical Design Document**

**Name:** Vanessa Belous

**Date Created:** October 26th, 2025

**Program Description:**

This program lets teachers store their student data in a csv file. A second program then prints the stored information in a formatted way.

**Functions used in the Program (listed in the order they’re called in):**

1. **Function Name:** write\_csv(total)

**Description:** This function prompts the user for their data, then writes that data to a csv file.

**Parameters:** total = the number of students a teacher has data for

**Variables:**

1. filename = the csv file name, in this case ‘grades.csv’
2. header = the header row that contains the column titles
3. csv\_writer = the writer object for the created csv file
4. counter = keeps track of how many students’ data has been written to the file
5. first\_name, last\_name = temporarily stores input for a student’s name
6. exam1, exam2, exam3 = temporarily stores input for a student’s three exam grades
7. student\_data = a list temporarily storing a student’s name and grades. The data gets written to the csv file through this list

**Logical Steps:**

1. the file name is defined
2. a header row is defined
3. the file is opened for writing in:
   1. a writer object is created
   2. the header is written to the file
   3. a counter variable is established
   4. a While loop:
      1. if statement checks whether or not the file has the amount of students the teacher has (the parameter of the function). Once the loop has iterated once for every student, the program exits the loop and closes the csv file.
         1. Print the student number so the teacher can keep track
         2. Input statements accept a student’s first and last name and exam grades, storing them temporarily in variables. Variable type is enforced: str for the names and int for the grades themselves
         3. These variables get assigned to a student\_data list. The list gets written as a row to the csv file (this list gets reassigned every time the loop runs, to write unique information to the csv file)
         4. Counter goes up one

**Returns:** nothing

1. **Function Name:** main()

**Description:** As the main function, this function calls the above function.

**Parameters:** None

**Variables:**

1. total\_students = stores how many students a teacher has, as an integer

**Logical Steps:**

1. A teacher (the user) is asked how many students they have
2. The amount of students is passed to the write\_csv(total) function, so that function knows how many rows to write to the csv file

**Returns:** nothing

**PROGRAM THAT READS CSV FILE (read\_csv.py)**

1. **Function Name:** main()

**Description:** This sole function of the program reads the csv file and displays the information in a tabular format.

**Parameters:** None

**Variables:**

1. csv\_reader = the reader object
2. formatted\_headers = stores the header; each header is formatted to be centered in a set width column
3. student\_number = stores the index number of a student, formatted
4. first\_name, last\_name = stores a student’s first and last name, formatted to be left justified in the column
5. exam\_1, exam\_2, exam\_3 = stores a student’s grades, formatted to be centered in the column

**Logical Steps:**

1. the file is opened to be read
2. a reader object is created
3. the header row is formatted, then printed
4. for very row in the csv file:
   1. all of the data is formatted, stored in temporary variables (the names are left justified, the grades and student number are centered)
   2. print the information stored in the temporary variables, in columns to align every row with the header row
5. close the file

**Returns:** Nothing

**Program Logical Steps:**

CSV\_Exercise.py

1. main() is called:
   1. how many students a teacher has data for is saved in a variable.
   2. that variable is passed to the function write\_csv(total)
2. after main() calls write\_csv(total):
   1. a csv file name and header row are defined
   2. the csv file is opened for writing
   3. the header row is written to the file
   4. a while loop runs as long as an if statement runs (based off how many records are written in the file):
      1. for every loop, a student’s information is accepted from the user, then written to the file as a row
      2. the counter goes up once per student written (this is how the function tracks how many rows of student data are written in the file)
      3. when all of the student data is recorded, the loop exits and the file closes.

read\_csv.py

1. main() is called:
   1. the grades.csv file is opened for reading
   2. the header row is formatted and printed
   3. for every row in the csv file, this function formats the data and prints it to be aligned with the header row
   4. once all rows are printed, the csv file is closed

**Link to Repository:** <https://github.com/VBelous1/COP2373/tree/master/Week%2010>

**Output Screenshot:**

CSV\_Exercise.py:

A screenshot of a computer

AI-generated content may be incorrect.

read\_csv.py:

A screenshot of a computer

AI-generated content may be incorrect.