

# CSE 8A Programming Assignment 8

Due Date: Tue. Dec 8th, 11:59 PM PT

## Need 1-on-1 Help?

You can get help by submitting a help ticket on the [Autograder!](#) Please read [Remote Tutoring Procedures For Students](#) for more instructions. Look at the [Course Calendar](#) to check when tutors/TAs are available! Look for "tutor hours" or "office hours" on the calendar. We expect that all students will need help at some point in time, so don't hesitate to reach out :)

## Learning Goals:

- Learn how to use lists of dictionaries
- Learn how to use dictionaries as values within a dictionary
- Understand how to use data structures to organize and analyze data

## Logistics:

All information for the following can be found on our [course syllabus](#).

- **Pair programming** - This programming assignment can be done individually or with a partner. Make sure to read the [guide on pair programming](#).
- **Academic integrity** - Please adhere to all academic integrity guidelines on the syllabus.

## Submission:

- You will submit the following files to Gradescope:
  - **pa8.py** - Contains your code for Part 1
  - **Pa8-writeup.pdf** - Contains all written portions of Part 2
  - **Pa8-video.mp4** - Your recorded video
- You can find the template of the write-up [here](#)
- You are also required to complete the [weekly reflection](#)
- Instructions on how to submit can be found below at [submission instructions](#).

## Part 1: Implementation (6 Points)

Save all of the code in the python file named *pa8.py*

### STEP A: Downloading Starter Code

To get started, first download the starter code [here](#). Click on the dropdown at the top that says **PA8 Starter Code** and then download. This will download a zip file. Unzip it and you are good to go. You will only modify the *pa8.py* file. Please do **NOT** modify the contents of *CSE8ACSV.py* file.

**Note:** If you are not logged into Google, you will see a different interface. In this case, there will be a button on the top-right corner reading "Download All". Click that button.

**Important:** Do not change the original folder structure, that is, do **NOT** rename/move files around. All the 4 files should be in the same folder with the given names.

### STEP B: CSE8A CSV Documentation

You are provided a few functions that help with loading data from CSV (Comma Separated Values) files in *CSE8ACSV.py*. Take a look at **blm\_state.csv** and **tech\_diversity.csv** to get an idea of the data and the format. You will **not** have to call any of these functions, as they are already implemented in the starter code provided. Below are the functions provided.

#### **get\_blm\_data(filename)**

You are given a function that will convert a CSV file (Comma Separated Values) containing data related to Black Lives Matter protests by state into a **list of dictionaries**, formatted as following:

```
[{State: String, TotalProtests: int, ...},  
 {another state}, ...]
```

This function is already present in the starter code to load the data into the variable *blm\_protest\_data*. Each dictionary in the list contains the following information:

Field Name	Field Type	Description
State	String	This field is a String that denotes the name of one of the 50 states in the US.
BlackPop	float	This field is a float representing the total black population of the state
BlackPoverty	float	This field is a float representing the percentage of black people below the poverty line in the state.
AsianPop	floatSt	This field is a float representing the total asian population of the state

AsianPoverty	float	This field is a float representing the percentage of asian people below the poverty line in the state.
HispanicPop	float	This field is a float representing the total hispanic population of the state
HispanicPoverty	float	This field is a float representing the percentage of hispanic people below the poverty line in the state.
WhitePop	float	This field is a float representing the total white population of the state
WhitePoverty	float	This field is a float representing the percentage of white people below the poverty line in the state.
TotalProtests	float	This field is a float representing the total protests in the state.
TotalAttendance	float	This field is a float representing the total attendance of the protests in the state.

### **get\_diversity\_data(filename)**

You are given a function that will convert a CSV file (Comma Separated Values) containing data of diversity in top technology companies into a dictionary, formatted as following:

```
{"Nvidia": {"hispanic": 3.4, "white": 42.9, "black": 1.0, ...},  
"Pinterest": {"hispanic": 3.7, "white": 50.2, "black": 2.1, ...}, ...}
```

This function is already present in the starter code to load the data into the variable `tech_diversity_data`. Remember that dictionaries are unordered, so the data may not be in any particular order. Each dictionary mapped to a company within the outer dictionary contains the following information:

Field Name	Field Type	Description
hispanic	float	This field is a float representing the percentage of hispanic people employed at the company.
white	float	This field is a float representing the percentage of white people employed at the company.
black	float	This field is a float representing the percentage of black people employed at the company.
asian	float	This field is a float representing the percentage of asian people employed at the company.
total_female	float	This field is a float representing the percentage of females employed at the company.

white_female	float	This field is a float representing the percentage of white females employed at the company.
asian_female	float	This field is a float representing the percentage of asian females employed at the company.
black_female	float	This field is a float representing the percentage of black females employed at the company.
hispanic_female	float	This field is a float representing the percentage of hispanic females employed at the company.
women_of_color	float	This field is a float representing the percentage of women of color employed at the company.
underrepresented_minorities_female	float	This field is a float representing the percentage of underrepresented female minorities employed at the company.
people_of_color	float	This field is a float representing the percentage of people of color employed at the company.
underrepresented_minorities	float	This field is a float representing the percentage of underrepresented minorities employed at the company.

### Part 1.1: Total BLM Protests (2 Points)

We are interested in calculating the total number of Black Lives Matter protests among a certain group of states. The data is already present in the starter code as a **list of dictionaries**, named `blm_protest_data`. Write a function that returns the total number of protests that took place in the list of states provided as the input parameter. Refer to the documentation [here](#) to review the structure of the list of dictionaries.

<b>Function Name:</b> <code>total_blm_protests</code>
<b>Parameter:</b> <code>states</code> - A list of strings denoting state names
<b>Return:</b> Returns an int representing the total number of protests in the list of state names (note that all <b>values</b> in the dictionaries are <b>floats</b> )
<b>Description:</b> This function uses the list of dictionaries <code>blm_protest_data</code> to obtain the total number of protests in the list of states.
<b>Examples:</b> <code>states = ["California", "Texas"]</code> Returns: 110

```
states = []
```

Returns: 0

```
states = ["Alaska", "New Mexico", "Arkansas", "New York"]
```

Returns: 72

## Part 1.2: Protest Attendance (2 Points)

We are now interested in finding out which states had a percentage of the population attend protests higher than a given threshold. Define a function to return a list of states that had a higher percent attendance than the threshold. For example, if 500 people attended protests in a state, and there were a total of 50000 people in the state, then the percentage of attendance for the state would be 1%. The same list of dictionaries, `blm_protest_data`, will be used for this problem. Refer to the documentation [here](#) to review the structure of the list of dictionaries. The total populations of states can be found in the variable `state_populations`, which is a dictionary of the following format:

```
{"California": 29044234, "Washington": 3180145, ...}
```

**Function Name:** *protest\_attendance*

**Parameters:**

`threshold` - a float that represents the *minimum* threshold for the percentage of the state's population that attended Black Lives Matter protests. `threshold` can be a value between 0 to 100 (both inclusive), but will only have realistic values between 0 and 1 (both inclusive), since in every state only less than 1% of the population attended the protests.

**Return:** The function should return a list of states that had a **higher** percentage of its population attend protests than the given threshold.

**Description:** This function calculates the percentage of people in each state that attended the Black Lives Matter protests, and returns a list of all states with a percentage attendance above the `threshold`

**Examples:**

```
threshold = 0.3
```

Returns: ['District of Columbia', 'Maryland', 'Minnesota', 'Missouri', 'New York']

```
threshold = 0.5
```

Returns: ['District of Columbia']

## Part 1.3: Diversity in Tech (2 Points)

For this problem, you will be using data on diversity in top technology companies. Use the dictionary `tech_diversity_data` to access the information. Refer to the documentation [here](#) to review the structure of the dictionary.

You would like to analyze the nested dictionary (i.e., dictionary of dictionaries) to find the companies that have a lower percentage of a given group of people than the given threshold, and return a dictionary of the companies and the percentage associated with the group at the company. For example, if `threshold = 50` and `field = 'total_female'`, then the returning dictionary should contain all of the companies with the percentage of total female employees **less than** 50%. Define the following function:

<b>Function Name:</b> <code>diversity_in_tech</code>
<b>Parameters:</b> <code>threshold</code> - a float that represents the <i>upper threshold</i> of the percentage of the field in the companies to be returned <code>field</code> - a String that represents which category within the dataset to analyze
<b>Return:</b> The function should return a dictionary with keys as Strings of company names, and values as floats representing the total percentage of people in that category at the company.
<b>Description:</b> This function should analyze the data in the dictionary of dictionaries to find companies that have a percentage of employees belonging to <code>field</code> <b>strictly below</b> the <code>threshold</code> passed in, and return a dictionary of these companies. The dictionary should be formatted as <code>{"company 1": float, "company 2": float, ... }</code> .
<b>Examples:</b> <code>threshold = 2</code> <code>field = 'black'</code> <b>Returns:</b> <code>{'Adobe': 1.6, '23andMe': 1.7, 'MobileIron': 1.8, 'Nvidia': 1.0}</code>  <code>threshold = 7</code> <code>field = 'asian_female'</code> <b>Returns:</b> <code>{'Apple': 5.9, 'Lyft': 6.3, 'HP Inc.': 4.4, 'HPE': 4.0, 'View': 6.5}</code>

## Part 1.4: Star Points (Optional)

You may have noticed that we didn't use all of the different fields in the datasets. For this assignment's star points, create a function that does some meaningful data analysis on one of the datasets provided. You are free to make it as complex or simple as you want, but the function must return a dictionary or a list of dictionaries. This question is completely open-ended, be creative!

## Part 2: Write up (3 Points)

You must report how you tested your code as well as answer a few short questions in **Pa8-writeup.pdf**. A [template](#) has been provided. See [submission instructions](#) below on how to make a copy of this template to your own drive. In particular, you must provide:

### A. Report Bugs and Issues

Include in your report any known bugs or issues with your program.

### B. Questions

Answer the following questions:

1. Explain similarities and differences between **lists of dictionaries** and **dictionaries of dictionaries**.
2. You are given the following dictionary:

```
movies =  
{  
    "Toy Story" : {"Rotten Tomatoes": 100, "IMDb": 8.3, "Rating": "G"},  
    "Finding Nemo" : {"Rotten Tomatoes": 99, "IMDb": 8.1, "Rating": "G"},  
    "Coraline" : {"Rotten Tomatoes": 90, "IMDb": 7.7, "Rating": "PG"},  
    "Coco": {"Rotten Tomatoes": 97, "IMDb": 8.4, "Rating": "PG"}  
}
```

A student has written the following statement to get all of the IMDb ratings into a list:

```
imdb_list = movies['IMDb']
```

Explain why this code is incorrect, and how you could write code to achieve the goal to create a list of all the IMDb ratings (i.e., `imdb_list` should contain the list with values `[8.3, 8.1, 7.7, 8.4]`).

### C. Star Point (Optional)

If you attempted the star point question, explain what your function returns, and what meaningful analysis it does.

## Part 3: Video (2 Points)

For this part, you will create a video recording explaining the code you have written. Your video should answer the questions below. If you are working with a partner, both partners should be in the video and each partner should have some speaking point. Example of following code execution: [https://youtu.be/7uw\\_Vi\\_F-dY](https://youtu.be/7uw_Vi_F-dY)

1. Follow the execution of the function `protest_attendance`, line by line, with the input

```
threshold = 0.25
```

2. Follow the execution of the function *diversity\_in\_tech*, line by line, with the input

```
threshold = 12  
field = 'women_of_color'
```

The following things will be checked in your video while grading:

1. The student(s) **code is clearly visible** in the video. [Hint: Increase your font size to 18  
Windows: "Options" → "Configure Idle" → Size  
Mac: "IDLE"(in top menu bar) → "Preferences" → "Fonts/Tabs" → "Size" ]
2. The student(s) clearly **answer(s) the questions**.
  - a. Students submitting individually answer both questions.
  - b. Students submitting in pairs each answer one of the questions.
3. Video is within **time limit** (max: **3 mins**)

## Part 4: Weekly Reflection (1 Point)

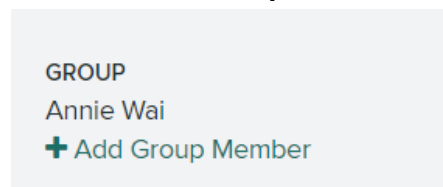
Fill out the reflection form ([link here](#)). This weekly reflection form is not optional, it counts towards 1 point of your assignment. All students have to **individually** submit their own weekly reflection regardless if you're working with a partner. Weekly reflections are not due when the PA is due. You may submit your weekly reflection for PA8 anytime before **11:59pm on Friday Dec 11th**.



# Submission Instructions

**Read all instructions carefully before submitting.**

- You will need to submit **pa8.py**, **Pa8-writeup.pdf**, and **Pa8-video.mp4** on Gradescope, and fill out the weekly reflection.
- To copy the writeup template and export as a pdf:
  - Click [here](#) to see a copy of the assignment format.
  - Click on "File" -> "Make a copy", and you will get a local copy of this Google Doc.
  - Fill in the Google Doc, making sure you keep the headings in about the same places.
  - Once you are done, in Google Docs, click on "File"-> "Download" -> "PDF Document", which will export it to a pdf.
- To record a video on zoom:
  - Start a zoom video meeting.
  - Choose "Join with Computer Audio"
  - If you're working with a partner, invite your partner into the meeting.
  - Share your computer's screen using "Share Screen"
  - Show your code on your computer's screen
  - Click "Record" > "Record on your computer".
  - Answer the required questions.
  - Once you are done, click "stop recording" and "End meeting".
  - Save your video file on your computer and name it as Pa8-video.mp4
- Sign into [Gradescope](#) and submit all files to PA8. You should be able to drag and drop multiple files into the upload files window. Ask a teaching staff for help if you are unsure whether you've submitted properly.
- If you are working with a partner, **only one member will need to submit the files** in Gradescope. Do not both submit the files individually to Gradescope. It will be your responsibility to ensure both members are added in Gradescope.
- To add a group member on Gradescope:
  - First submit all files to PA8.
  - This should take you to your submissions page. Otherwise, you can view your submission by clicking on the assignment.
  - Click on **"Add Group Member"** on the top right under your name.



- Confirm you have added your partner. You should see both you and your partner's name under "Group" in the top right after submitting.
- You may submit multiple times until the deadline. We will be grading only your latest submission. So, please make sure that your latest submission is your best version!