

## Computer Project #06

(Sample output updated on Oct. 17)

### Assignment Overview

This assignment focuses on the implementation of Python programs to read files and process data by using lists and functions.

It is worth 45 points (4.5% of course grade) and must be completed no later than 11:59 PM on Monday, October 24.

### Assignment Deliverable

The deliverable for this assignment is the following file:

`proj06.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline.

### Assignment Background

Executions for exceptionally severe crime, usually murder, have been U.S. policy for years. Recently some have questioned its fairness, usually with respect to race. That is, are people of some races more likely than others to be executed? The state of Texas has one of the highest execution rates in the U.S. and posted a spreadsheet of data about 500 recent executions. It is `Texas_death_row.csv`. The spreadsheet is large with details from characteristics of the convicted and the victim to the final statement before execution. As is often the case with such data there are many gaps.

Your assignment is to investigate **and answer** these four questions with respect to Texas (answers go in comments at top):

1. Do minorities make up a higher percentage of whites in executions?
2. Do minorities killing whites make up a higher or lower percentage of executions than whites killing minorities?
3. Do men killing women make up a higher or lower percentage of executions than women killing men?
4. Do men make up a higher or lower percentage of executions than women?

You may make the following assumptions:

1. All executions in the spreadsheet are for murder.
2. Any field that is missing data is blank or labeled with the string: 'Not Available'
3. Data on race has only these labels: white, black, Hispanic, or other (or blank or 'Not Available')
4. A minority's race is either black or Hispanic (the predominant minorities in Texas).
5. Ignore the co-defender, if any.
6. Ignore the data of any convict whose race is blank, other, or 'Not Available'
7. When counting victim data use the following protocol so we are all counting the same when there are multiple victims:

- a. For minority-on-white, count as white, if any victims are white.
- b. For male-on-female, count as female, if any victims are female
- c. For white-on-minority, count as minority, if any victims are minority.

Other useful information about the data:

1. The murderer has separate fields for race and gender (labeled “Race” and “Gender”).
2. The victim data on race and gender is combined into the same entry and if there is more than one victim the data on all victims is combined into the same entry with a wide variety of format and often missing data. E.g. “Hispanic male, Hispanic female”, “1 White/Male; 2 Unknown/Females”, “2 adult males”. The field is labeled “Race and Gender of Victim”.
3. We only consider the three fields labeled: “Race”, “Gender”, and “Race and Gender of Victim”

## Assignment Specifications

1. The program must provide following functions to extract some statistics.
  - a) `open_file()` prompts the user to enter a file name. The program will try to open the data file. Appropriate error message should be shown if the data file cannot be opened. This function will loop until it receives proper input and successfully opens the file. It returns a file pointer. Use this function from previous projects.
  - b) `read_file()` calls the `open_file()` function and uses the returned file pointer to read the data file. This function returns a list of your choosing containing data you need for other parts of this project. We suggest creating and returning a list of tuples.

```
T = (r,g,v) # create a tuple T
L.append(T) # append tuple onto list L
```

Important: use the csv package because some entries have commas so that splitting on commas doesn't work. The `csv.reader` function takes a file pointer as an argument.

Remember to place `import csv` at the top of your program. Note that when you iterate through what `csv.reader` returns you are iterating through a list.

```
fp = open_file() # open file as usual
csv_fp = csv.reader(fp) # invoke reader
for L in csv_fp: # iterate
    print(L) # print list
```

You will create a list rather than print a list.

- c) You may use extra functions, if you wish.
2. The program should read the file only once.
3. **ANSWER THE FOUR QUESTIONS in comments at the top of your program.** Keep your answers brief, e.g. one sentence is usually sufficient, but **you must** cite the program output to support your conclusion. For example: “3. An execution for male-on-female murder is more likely than female-on-male because there were 65 male-on-female murders vs. 1 female-on-male murders whereas females and males make up roughly equal percentages of the overall population.”

## Assignment Notes

1. Items 1-9 of the Coding Standard will be enforced for this project.
2. Note that data are separated using spaces. You can use the list method `.split()` to split the line into a list of data.

3. It is much easier to convert the input data to int and float when the program reads the file and creates the list.
4. To create a list `L` of data begin with an empty list (e.g. `L = []` before the loop begins) and within the loop append to the list one item at a time, e.g. `L.append(item)`. The data item you append may be a collection such as a tuple or another list.

### Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step may be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.
- Construct the program one function at a time—testing before moving on.
- Use the **handin system** to turn in the first version of your solution. Cycle through the steps to incrementally develop your program:
  - Edit your program to add new capabilities.
  - Run the program and fix any errors.
  - Use the **handin system** to submit the current version of your solution.
- Be sure to log out when you leave the room, if you're working in a public lab.

### Sample Output

```
Enter a file name: Texas_death_row.csv
=====
White vs. Minority
N = 498
White: 225
Black: 187
Hispanic: 86
Minority = Black + Hispanic: 273
=====
Male vs. Female
N = 119
Male = 118
Female: 1
=====
Race difference between perpetrator and victim.
N = 117
Minority-on-white: 28
Male-on-female: 65
Female-on-male: 0
White-on-minority: 7
```

=====

## Educational Research

**When you have completed the project insert the 5-line comment specified below.**

For each of the following statements, please respond with how much they apply to your experience completing the programming project, on the following scale:

1 = Strongly disagree / Not true of me at all

2

3

4 = Neither agree nor disagree / Somewhat true of me

5

6

7 = Strongly agree / Extremely true of me

*\*\*\*Please note that your responses to these questions will not affect your project grade, so please answer as honestly as possible.\*\*\**

**Q1: Upon completing the project, I felt proud/accomplished**

**Q2: While working on the project, I often felt frustrated/annoyed**

**Q3: While working on the project, I felt inadequate/stupid**

**Q4: Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this course.**

Please insert your answers into the bottom of your project program as a comment, formatted exactly as follows (so we can write a program to extract them).

# Questions

# Q1: 5

# Q2: 3

# Q3: 4

# Q4: 6