Assignment #4

Assignment Overview

In this assignment you will practice with lists, tuples, functions, and CSV file manipulation to perform basic queries and produce simple plots related to **vehicle fuel consumption**.

Background

The US Environment Protection Agency (EPA) publishes **fuel economy data** that "are the result of vehicle testing done at the Environmental Protection Agency's National Vehicle and Fuel Emissions Laboratory in Ann Arbor, Michigan, and by vehicle manufacturers with oversight by EPA."
See: https://www.fueleconomy.gov/FEG/download.shtml

They also provide interactive tools to explore some of the data and plot trends (over time):
https://www.epa.gov/automotive-trends/about-automotive-trends-data

Project Specification

In this assignment you will implement a simple solution for querying the EPA fuel consumption data and producing customizable plots of relevant trends over time.

Your program ("**baseline solution**") will:

1. Print a brief (2-5 lines) message explaining the purpose of this "app".
2. Prompt for:
    - Main option: (1) Mileage info or (2) Trend plot
3. If option == 1:
    - Prompt user to input the **MPG interval** they are interested in.
    - Find the vehicles that are within that range and store their information in appropriate data structures. The data file for this part is called epaData2015.csv
    - Display information (make, model) about vehicles that fall withing that range (*similar to the output of code listing 7.15 in the textbook*).
4. If option == 2:
    - Prompt user to input the measure that they are interested in.
        - At a minimum, offer 3 options: (H)ighway MPG, (C)ity MPG and (O)verall MPG
    - Prompt user to input the preferred way to produce the plot:
        - *(D)isplay on screen* or *save to (F)ile*.
    - Gather the necessary data from the CSV file and store it in appropriate data structures. The data file for this part is called epaData2020.csv
    - Prepare and display/save the plot using pylab (*see code listing 7.17 in the textbook*).

Requirements

You are required to:

1. Use **import csv** and **csv.reader()** to read the CSV files.
2. Design and implement a function that stores MPG information (for option 1) in an appropriate data structure (e.g., a list of tuples).
3. Design and implement *another* function that stores MPG information (for option 2) in an appropriate data structure (e.g., two lists containing the x and y coordinates needed by the plot function).

Deliverables

You must submit (via Canvas):

- The file **a4_FAUusername.py** (where "**FAUusername**" is your FAU username)
    - This is your source code solution; be sure to include your name, date, assignment number and comments describing your code.
- A **README.md** file with "project notes" (describing what my TA and I cannot see by looking at your source code and/or running your program).
    - Examples: design decisions, documented limitations, future improvements, etc.
- **Screenshots of the results** produced by your code (make sure to show input and output for **at least four runs, two per option**).

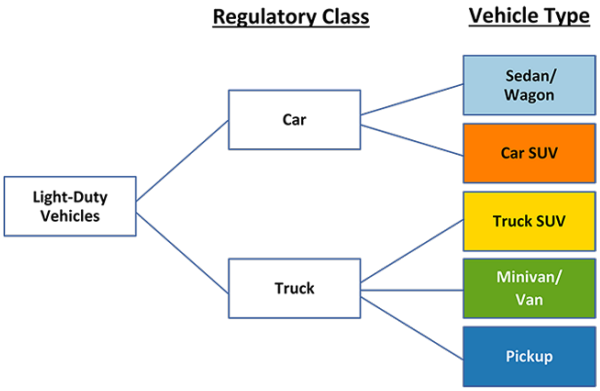
Notes and Hints:

- Follow the "cardinal rules" of programming in Python (as per the textbook).
- Start by breaking the program down into parts and solve smaller problems before producing the final solution.
- Try to handle special cases and prevent runtime errors to the best of your knowledge.
- Don't overdo it! You might want to try some of the bonus points options below but should refrain from going beyond the scope of the course so far.
    - Things to avoid (as cool as they might be) include using the Python "data science stack" (pandas, matplotlib, seaborn, numpy, scipy, scikit-learn).

Bonus opportunities:

If you successfully implement one or more of the bonus options below **without breaking the baseline solution** (*) you will earn extra points (max 10% each):

1. **Data cleanup**
    - As you will soon be able to tell, the CSV files are messy (inconsistent capitalization, incomplete data, etc.).
    - If you implement (and document) an **elegant** solution for cleaning up the CSV (and demonstrate it using another file from the EPA website), you will earn up to extra 10 points.
2. **Nicer plots**
    - The EPA interactive website allows multiple curves to be plotted (using different colors) for different, e.g., for different vehicle types.
    - If you implement an improved solution that allows multiple vehicle types / regulatory classes[1] to be taken into account and produce richer plots, you will earn up to extra 10 points.



[1] The dataset used for 'option 2' is limited to the 'Car' regulatory class.

(*) A common mistake students make is to attempt bonus items before producing a "perfect 100" baseline solution.
Make no mistake: if your baseline solution doesn't meet **all** grading criteria (see rubric on Canvas), you are not eligible for bonus points. Period.