INF 405 & 505, Fall 2022

Individual Programming Assignment Two

Due: Please refer to Blackboard course calendar for all due dates

Description: Reading in and providing some details about tuition cost data for US States. Also using this data to make predictions about future prediction costs

This program will use as its primary resource a file of tuition cost data for US States for recent years. The input file is `us_avg_tuition.csv` and is available in the Blackboard folder for this assignment.

Specific Program Behavior

1. The program will read in the data file. This will give it the data it needs to answer the different types of user queries.
2. The program will then operate in a loop. In the loop, the user can specify three different types of queries.
   a. Whole US data.
   b. Data for a specified US State
   c. Exit the program

The queries are detailed as follows:

- Specified State Data:
  For the specified State, please report:
  - The overall average tuition cost for the years covered.
  - The average rise between years for all the years covered.
  - A *prediction* for the next one year's tuition based upon:
    - The tuition rise for the last year,
    - The average tuition rise for the years covered.
  - A prediction for the tuition raise two years out (i.e. year after next) based upon:
    - The tuition rise for the last year,
    - The average of the tuition rise for the last two years
    - The average tuition rise for all the years covered.
- Whole US data:
  - Same calculations as for a State (see above) but averaging all the States to create US data.
- Exit
  - If the user requests the program exit, the program terminates.
  - Otherwise it goes back into the loop prompting the user for its query.

Requirements:

- You must organize this as an object-oriented program. That is, your data must be defined as classes, and objects based upon those classes used.
- You may not use packages such as numpy, pandas, *etc*. You may use simple mechanisms, such as csv to read in data.
- The program must be organized using multiple code files (e.g. modules, packages). A single file program is not permitted.
- There will be an intermediate deadline for this assignment (due date on Blackboard). On that date, you must turn in the work you have done to that point. This is to establish that the program has been organized, planned, and thought out. This is in contrast to doing it at the last minute. The things you should emphasize for the intermediate deadline are:
    - The classes for your data
    - The overall logical architecture of the program.

  This submission will not be graded for code correctness, execution, etc. The purpose is to encourage you to plan the assignment. The intermediate submission will be worth 20% of the overall grade for IPA2.

Hints:

- The individual underlying problems here are not difficult one-by-one. The main complexity is that there is a lot going on. You will have to be organized, especially in your initial design of the program to keep things clear.

Notes

- The program should use comments to describe its overall operation and to identify the specific operation of significant lines in the code.
- This is an individual assignment, so please do your own work. Except as explicitly noted, no collaborating with others, looking up or soliciting answers from the internet or other sources.
- If you have difficulties, please use the Discussion Board set up for the assignment on Blackboard, ask questions, or see the instructor.
- Please follow the submission details given below to get credit for this program.

Submission

You must submit:

1. Your source language file(s) containing your program via Blackboard.

Grading

The program is graded based upon *specifications*. Specifications are specific aspects of the assignment that are important. Each specification has a certain number of points associated with it. Each specification is all-or-none (that is, you get either zero or full credit for each specification). The total of your specification scores is your score for the program (out of 100)

The Specifications

Intermediate Submission:

- Represents an overall design (the logical architecture) of the program (5 points)
- Defines the Object-oriented architecture of the program
  - Classes that represent the major data elements necessary (3)
  - Object instances of these classes (2)
- Is in-depth enough to represent a true start on the program
  - All of the major elements of the program are represented (3)
  - There is enough detail to be credible (2)
- Is documented sufficiently for a reader to understand
  - Overall and inline documentation (3)
  - Meaningful class, etc. names (2)

Final Submission:
Programming

- Uses valid basic Python v3 (no PANDAS or other libraries other than CSV) (4 points)
- The Code is Commented
  - Inline (2)
  - Overall (2)
- The program is designed using Object-oriented principles (i.e. classes and objects are the major way of organizing data in the program) (4)
- The program is implemented using multiple source files (2)

The Assignment

- Reads in the data file (4)
- Correctly processes the data file (4)
- Contains the main loop (4)
- Implements the Whole US Option
  - US: The average tuition rise (4)
  - US: The tuition rise for each year (4)
  - US: Correct one-year prediction
    - Overall correctness (3)
    - Last year-rise based (2)
    - Average rise based (2)
  - US: Correct two-years-out prediction
    - Overall correctness (3)
    - Last year-rise based (2)
    - Last two years rise based (2)
    - Average rise based (2)
- Implements the Specified State Option
  - State: Correctly identifies the specified State (4)

    - State: Correctly handles an invalid State name (4)
    - State: The average tuition rise (4)
    - State: Correct one-year prediction
        - Overall correctness (3)
        - Last year-rise based (2)
        - Average rise based (2)
    - State: Correct two-years-out prediction
        - Overall correctness (3)
        - Last year-rise based (2)
        - Average rise based (2)
- Implements the Exit Option (4)