

(CH9) Program Assignment Instructions

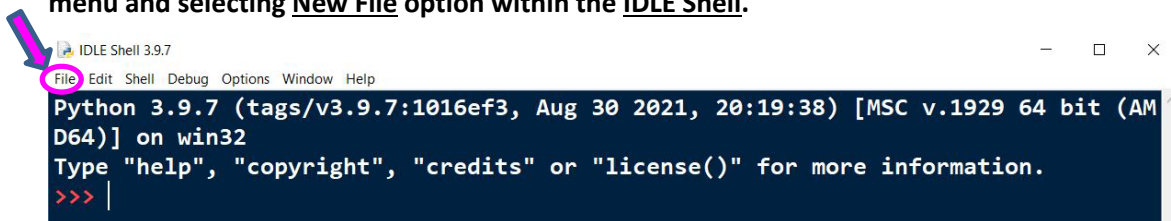
Last Changed: 9/16/2022 12:19 PM

Read and follow the directions below carefully and perform the steps in the order listed. You will be solving one program as instructed and turning in your work electronically via an uploaded file within Eagle Online/Canvas. Make sure and check your work prior to uploading the file.

*Note: Refer to **(SET) How to Download Install and Use Python IDLE (Windows User)** file (Page 8) and/or **Use Python IDLE Video** link within **Module 2** on how to create, enter, save, run, and submit a script (source code/program) file.*

Instructions

1. Using Python IDLE, create a New Empty **Script (Source Code/Program) File** in your working drive by clicking the **File** menu and selecting **New File** option within the **IDLE Shell**.



2. Save the new **Script (Source Code/Program) File** with the name **CH9LastFirst.py** by choosing **File** menu's **Save** option (*Note: Make sure to know where you saved the file in your working directory for future submission*)

(NOTE: Where **LastFirst** should be replaced with your actual **Lastname** and **Firstname**. For Example, if your name is Mary Smith then your file name should be named: **CH9SmithMary.py**. Reminder: You don't need to add the **.py** extension, IDLE will automatically add the extension)



3. You will develop a **Professor Evaluation Report** program (using objects/classes, lists, and loops) as described below:

Computer Science Department is evaluating its professors to see which professor has the highest rating according to student input. You will create an OOP with a **Rating** class consisting of professor's name and three ratings. The three ratings are used to evaluate **easiness**, **helpfulness**, and **clarity**. The value for each rating is in the range of 1 to 5, with 1 being the lowest and 5 being the highest). Further, you will calculate and display the professor's average rating based on the three ratings, his/her name, and the three ratings. You will further find and display the professor with the highest average rating (Refer to screenshot in step 6 below for an example display).

4. **Ensure your program meets all the following criteria:**
 - i. A class named **Rating** with four private data fields as described below:
 - a. **name**: a private string data field for professor's name
 - b. **easy**: a private int data field for rating professor's easiness
 - c. **helpful**: a private int data field for rating professor's helpfulness
 - d. **clear**: a private int data field for rating professor's clarity

- ii. Your **Rating** class should contain the following methods:
 - a. An initialization method (**__init__**) that creates a rating with the specified values for the four data fields (see example program screenshot below)
 - b. **getter** and **setter** methods for all four data fields
 - c. A method named **calcAvgRating ()** that returns the professor's average rating by dividing the sum of three ratings (**easy**, **helpful**, and **clear** data fields) by 3
- iii. In your **main ()** program, do the following (see example program screenshot below):
 - a. Create 3 **Rating** objects: **rate1**, **rate2**, and **rate3**
 - b. For the first object **rate1** assign name: **Mary**, easy: **5**, helpful: **5**, and clear: **5** and turn it on
 - c. For the second object **rate2** assign name: **Joe**, easy: **4**, helpful: **5**, and clear: **3** and turn it on
 - d. For the third object **rate3** assign name: **Ria**, easy: **4**, helpful: **4**, and clear: **5** and turn it on
 - e. Define a 3-element rating list named **rateList** and assign with the 3 objects created
 - f. Call a generic function **displayRating(rateList)**: Define this function to display each professor's name, his/her three ratings and the average rating (making sure to format average to two decimal places). Refer to screenshot in step 6 below for an example display
 - g. Call a generic function **hiIndex = findHiIndex (rateList)**: Define this function to return the index of the highest average rating
 - h. Call a generic function **displayHiRating (rateList, hiIndex)**: to display the professor's name and his/her highest average rating (making sure to format average to two decimal places). Refer to screenshot in step 6 below for an example display
- iv. **A top comment block** with your full name, date, program topic(s), and the program description (this should clearly explain the program logic/design in your own words).
- v. Define appropriate/descriptive **CONSTANTS, Variables, and method/function names** in your program (refer to EXAMPLE PROGRAMS listed in Canvas modules and example program screenshot below)
- vi. Have enough **documentation for understandability** of your program by including **comments** appropriately. Include enough comments explaining the logic of the program where required (refer to EXAMPLE PROGRAMS listed in Canvas modules)
- vii. Have proper **indentation and line spacing** for readability of your program (refer to EXAMPLE PROGRAMS listed in Canvas modules and example program screenshot below)
- viii. It is imperative that your output **looks professional!** That is, no typos! Use capital letters at the beginning of a sentence/phrase. Use punctuation when appropriate, etc.

Below screenshots help to create some of the steps listed above:

```

1 '''
2 Name: Your full name here...
3 Date: Program creation date here...
4 Program Topic: Objects/Classes, Lists, and Loops
5 Program Description: Clearly explain the program logic/design in your own words here...
6 '''
7
8 # CONSTANTS
9 TITLE = "Welcome to Professor Evaluation Report Program!\n"
10 NUM_RATING = 3
11 COL_TITLE = "\nProf.Name    Easiness    Helpfulness    Clarity    AvgRating\n"
12 LINE = '- '*len(COL_TITLE)
13
14 # Rating class definition
15 class Rating:
16     def __init__(self, name, easy, helpful, clear):
  
```

The screenshot shows a Python IDE window titled "CH9RaoSuma.py - C:\COSC1437Py\Fa21Examples\CH9RaoSuma.py (3.9.7)". The code is written in a dark-themed editor. A pink bracket on the left groups lines 1 through 6, with a pink arrow pointing to a yellow box labeled "Step iv". Another pink bracket on the left groups lines 8 through 12, with a pink arrow pointing to a yellow box labeled "Step v". A pink arrow points from a yellow box labeled "Step i" to line 15. Another pink arrow points from a yellow box labeled "Step ii (a)" to line 16. The code includes a top comment block (lines 1-6), constants (lines 8-12), and the start of a Rating class definition (lines 14-16).

Instructions

<COMPLETE REMAINING PROGRAM HERE...>

```
60
61 # main function definition
62 def main():
63     rate1 = Rating("Mary", 5, 5, 5)
64     rate2 = Rating("Joe", 4, 5, 3)
65     rate3 = Rating("Ria", 4, 4, 5)
66
67     rateList = [rate1, rate2, rate3]
68
69     displayRating(rateList)
70     hiIndex = findHiIndex(rateList)
71     displayHiRating(rateList, hiIndex)
72
73 # main function call
74 main()
```

Step iii (a to d)

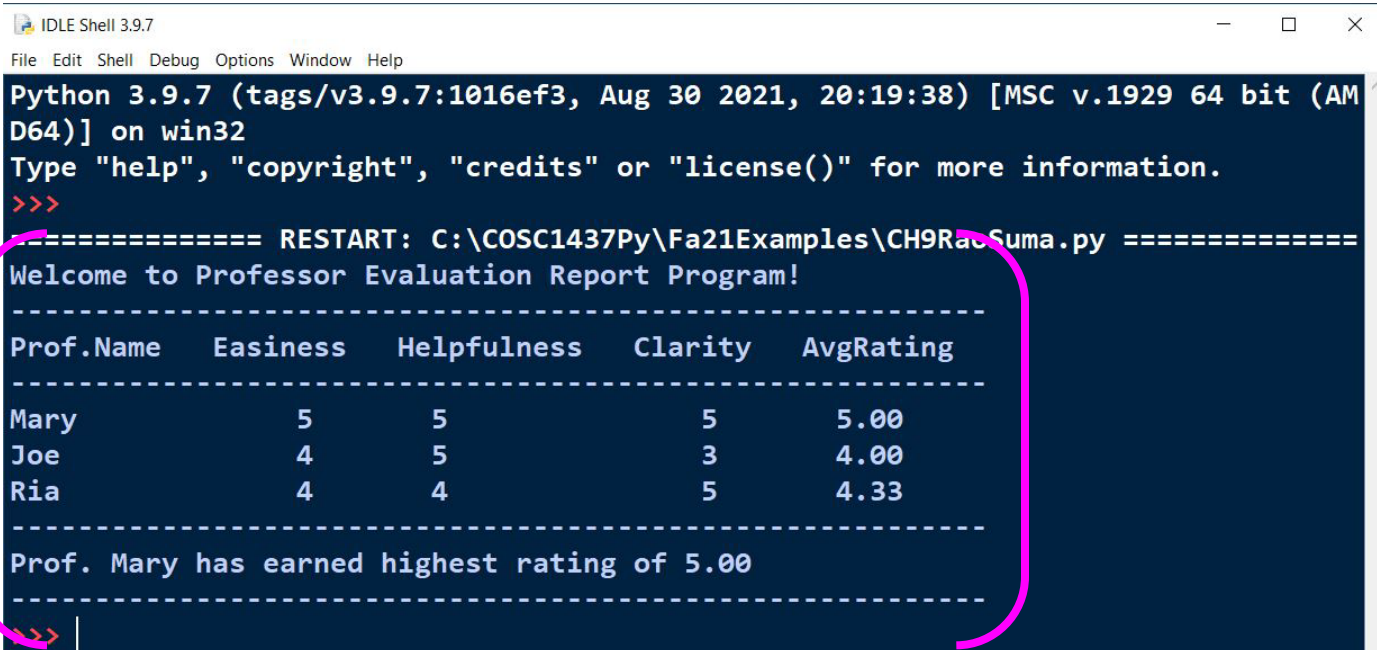
Step iii (e)

Step iii (f, g, h)

5. After completing your program as instructed, make sure to run your program by clicking the Run menu and selecting the Run Module (or F5 function key) option within the Script (Source Code/Program) file.



6. Your Script (source code/program) File output should be displayed in the IDLE Shell as shown in the example output screenshot below:



Instructions

7. You may now proceed to Program Assignment INSTRUCTIONS and UPLOAD link within this module and follow the instructions in the link or follow the below steps to Upload the script (source/program) file you created in step 2 above:
 - a. Choose the **Start Assignment** button,
 - b. Choose **File Upload** tab,
 - c. Choose **Browse** to locate your script (source/program) file to add,
 - d. Choose **Submit Assignment** to complete file upload.

**NOTE: ONE OF THE COMMON MISTAKES IS THAT STUDENTS ENTER PYTHON COMMANDS/STATEMENTS IN THE "IDLE SHELL" DIRECTLY AND SAVE THE RESULTS TO A FILE AND SUBMIT WHICH IS INCORRECT!!!
INSTEAD...**

YOU SHOULD FOLLOW THE ABOVE STEPS TO CREATE A NEW SCRIPT (SOURCE CODE/PROGRAM) FILE FROM THE IDLE SHELL, SAVE THE FILE, ENTER PYTHON STATEMENTS (PROGRAM) INTO THE FILE, RUN YOUR PROGRAM, AND SUMBIT THAT SCRIPT (SOURCE CODE/PROGRAM) FILE AND NOT THE OUTPUT OF THE IDLE SHELL!!!