CS 1026A Assignment 1: Inflation Rate Due: October 5

Introduction

Inflation is the rate of increase in prices over a time period. The personal inflation rate refers to the rate at which inflation increased for a household. The calculation of the personal inflation rate requires as input the expenses for each expenditure category (e.g., food, tuition, rent, transportation, entertainment) for a specific month of the current year and for the same month in the previous year. Individuals may have different expenditure categories. For example, not everyone pays tuition.

The expenses for each category should be summed up for both years. The sum of all expenses of the previous year is subtracted from the current year. The result is then divided by the monthly expenses of the current year.

$$Inflation\ rate = \frac{\text{total expenses of the current year} - \text{ total expenses of the previous year}}{\text{total expenses of the current year}} \times 100\%$$

For example, let us assume that you have these categories of expenses: food, rent and transportation. The following table represents expenses for food, rent and transportation for August 2022 and August 2021.

Year	**Food**	**Rent**	**Transportation**	**Sum of Expenses**
2022 (current)	200	1000	500	1800
2021 (previous)	100	800	300	1200

The inflation rate is

(1800-1200)/1800 = 600/1800 = 0.33 which represents an inflation rate of 33%.

In this assignment, you will get practice with:

- Basic Python programming constructs
- Expressions, decisions, loops
- Getting input from users
- Algorithm development and testing; designing test cases
- Following program specifications

Files

For this assignment, you must create one Python file: *Assign1.py*

When submitting your assignment on Gradescope, please submit Assign1.py only. Do not upload any other files.

Assign1.py

In this assignment, you will write a complete program in Python that computes the personal inflation rate and determines if the calculated information rate is low, moderate, high or hyper. A low inflation rate is below 3%, a moderate is 3% or higher but less than 5%, high inflation is more than 5% but less than 10% and any inflation rate at 10% or higher is hyper.

Your program is expected to prompt the user for input and compute results. Your program will make use of expressions, decisions and input/output in Python. You should name your program Assign1.py. Your program should strictly adhere to the Functional Specifications (below) and Non-Functional Specifications.

Functional Specifications

Your program will compute the person's inflation rate as described above. The program should prompt the user for the number of expense categories. This number is used to determine the number of times that the user is asked for category and expense. It will then print the inflation rate and if the inflation rate is low, moderate, or high.

<u>The program should prompt for these values in the order described.</u> An example of the program's input is:

```
Please enter the year that you want to calculate the personal interest
rate for: 2022

Please enter the number of expenditure categories: 2

Please enter expenses for previous year: 100

Please enter expenses for year of interest: 100

Please enter expenses for previous year: 200
Please enter expenses for year of interest: 300

Personal inflation rate for 2022 is 25.0%

Type of Inflation: Hyper
```

The output consists of a two lines. The first line is the inflation rate and the second is the level. The first line **MUST** be identified with phrase "Personal inflation rate:" followed by the inflation rate. The second line must be identified with the phrase "Type of Inflation" followed by the type of inflation. An example output is below:

Personal inflation rate for 2022 is 25.0%

Type of Inflation: Hyper

Finally, an automated testing program will run a number of test cases against your program. Some examples of output and test cases are presented below; these are NOT comprehensive - you should create your own test cases and thoroughly test your program.

Example Executions and Output

# Example 1

```
Please enter the year that you want to calculate the personal interest
rate for: 2022

Please enter the number of expenditure categories: 2

Please enter expenses for previous year: 100

Please enter expenses for year of interest: 100

Please enter expenses for previous year: 200

Please enter expenses for year of interest: 300

Personal inflation rate for 2022 is 25.0%

Type of Inflation: Hyper


       Process finished with exit code 0
```

# Example 2

```
Please enter the year that you want to calculate the personal interest
rate for: 2021

Please enter the number of expenditure categories: 3

Please enter expenses for previous year: 100

Please enter expenses for year of interest: 100

Please enter expenses for previous year: 200

Please enter expenses for year of interest: 200

Please enter expenses for previous year: 300

Please enter expenses for year of interest: 300

Personal inflation rate for 2021 is 0.0%

Type of Inflation: Low


     Process finished with exit code 0
```

Non-Functional Specifications

1. The program should strictly adhere to the input and output requirements, particularly the order of the input and the labeling of the output cost.
2. The program should include brief comments in your code identifying yourself, describing the program, and describing key portions of the code.
3. Assignments are to be done individually and must be your own work. Software may be used to detect academic dishonesty (cheating).
4. Use Python coding conventions and good programming techniques. For example:
    a. Meaningful variable names
    b. Conventions for naming variables and constants
    c. Use of constants where appropriate
    d. Readability, indentation, and consistency

5. The name of the file you submit must be **Assign1.py**

Make sure you develop your code with **Python 3.9** as the interpreter. Failure to do so may result in the testing program failing.

Tips and Guidelines

- Variables should be named in lowercase for single words and camel case for multiple words, i.e. infRate
- You can assume:
    - Expenses are floating-point numbers
    - All values entered are valid – numbers that are greater than 0. You do **not** need to check if the input is valid
- Add comments throughout your code to explain what each section of the code is doing and/or how it works

Rules

- Read and follow the instructions carefully.
- Only submit the Python file described in the Files section of this document.
- Submit the assignment on time. Late submissions will receive a late penalty of 10% per day (except where late coupons are used).
- Forgetting to submit a finished assignment is **not** a valid excuse for submitting late.
- Submissions must be done on Gradescope. They will not be accepted by email.
- You may re-submit your code as many times as you would like. Gradescope uses your last submission as the one for grading by default. There are no penalties for re-submitting. However, re-submissions that come in after the due date **will** be considered late and subject to penalties (or to the use of late coupons).
- Assignments will be run through a similarity checking software to check for code that looks very similar to that of other students. Sharing or copying code in any way is considered plagiarism and may result in a mark of zero (0) on the assignment and/or reported to the Dean's Office. Plagiarism is a serious offence. Work is to be done **individually**.

Submission

Due: Wednesday, October 5, 2022 at 11:55pm

You must submit the file (Assign1.py) to the Assignment 1 submission page on Gradescope. There are several tests that will automatically run when you upload your files. Some of the tests are visible to you so it will give you an idea of how well your program is working. However, we will also run some hidden tests so you won't see those tests nor the grade you get from those tests. It is recommended that you create your own test cases to check that the code is working properly for a multitude of different scenarios.

Assignments will not be accepted by email or any other form. They **must** submitted on Gradescope.

Marking Guidelines

The assignment will be marked as a combination of your auto-graded tests (both visible and hidden tests) and manual grading of your code logic, comments, formatting, style, etc. Below is a breakdown of the marks for this assignment:

[50 marks] Auto-graded Tests

[20 marks] Code logic and completeness

[10 marks] Comments

[10 marks] Code formatting

[10 marks] Meaningful and properly formatted variables

Total: 100 marks

The weight of this assignment is 5% of the course mark.