Assignment 2

INFSCI 0201 - Intermediate Programming with Python. Fall 2023

Due Date: November 10, 2023, before 11:59 PM

In this assignment, you will work with a class to test your understanding about overriding equality and hash methods. In addition, you will work on serializing objects using the JSON format.

Submission Guidelines:
In this assignment, you will work on a Jupyter Notebook shared in the lecture's repository:
https://github.com/cskamil/fall2023_0201/blob/main/Assignment%202/Assignment_2.ipynb

1- Download the Jupyter notebook on your local machine.
2- Copy it to your Assignment repository folder so that you can later submit your code through your own repository as you did in Assignment #1.
3- Follow the instructions shared in Part 1 and Part 2 below and complete the Jupyter Notebook.
4- Test your code with the provided tester code – The last cell in the provided notebook.
5- Commit your code to your own repository. Make sure you have pushed your changes so that I can see them for grading. Make sure that you can see your code on GitHub.com before you think you are done.
6- Submit your repository link through Canvas. Even if you use the same repository for Assignment #1, you need to perform this step.

Grading Policy
The tester code in the notebook has the grading policy integrated as comments. You will lose points for the parts that is not working as expected.

Part #1: Equality and Hash Methods

Consider the Professor class shared in the notebook that represents professors in a university. Override __eq__ and __hash__ methods to enable comparison and hashing based on the 'name' and 'staff_id' attributes.

Part #2: Serialization in Python using JSON strings

Consider the Professor class shared in the notebook. This time you need to implement a method called 'to_json' that returns all the instance properties as a JSON string.

Part #3: Serialization in Python using JSON files

In the final part, you will update the Professor class to include a method that serialize the objects to a JSON file and deserialize it from a given JSON file. Hint: You can use the method you have implemented in Part 2.