Alexis Harris

IT FDN 110A

March 15, 2025

Assignment 07

GitHub - https://github.com/ahar9/IntroToProg-Python-Mod07

Assignment 07

Introduction

Assignment 07 for Foundations of Python programming delves into separations of concern processes using classes, constructors, attributes, properties, and inheritance methods. This assignment fleshes out the uses of classes and methods to create instances and modifications for Object Oriented Programming (OOP). At the end, I modified Assignment07_starter.py to practice these important concepts.

Reviewing Module 07 Text

The Module07 text covers how to create classes to manage and organize data. While it may seem tedious, these practices help with scaling programs for complex logic and size. I learned that a statement (data statements, processing statements, presentation statements, looping, and conditional statements) can be sorted into functions to reuse. These functions can be sorted into classes and are then referred to as methods. Classes keep larger programs organized by grouping statements and methods.

Organizing classes is essential for programming. Classes can be organized into data classes, presentation classes, and processing classes. Data classes can use constructors and properties to initialize and modify class attributes. The presentation and processing classes can utilize objects with attributes from the data class to present the data and process the data from different data types. These concepts help programmers make their program user friendly and reusable.

While organizing classes, it's important to note how to utilize class inheritance. I learned in Python programming can overwrite inherited whimsical methods that passes to all objects: __init__() and __str__(). We can also pass classes methods and attributes, for example passing a Person class to a Student class. Class inheritance and overwriting methods help programs become reusable.

Assingment07.py

To implement the concepts from Module07, I referred to the lecture, the module labs, and demos. Below are some comparisons of the acceptance criteria to the starter script.

Comparing Acceptance criteria to starter python script:

- Updated script header
- Constants look consistent
- Variables look consistent between criteria and starter script

Classes

- Added class named Person and class named Student with description; noticed there is green comments with TODO indicating this instruction to add these classes.

Class Properties

- When inheriting properties between Person and Student class, I thought it best to use the first_name and last_name attributes from Person rather than make new attributes for student_first_name and student_last_name.

Class Methods:

- Modified method in the IO class to extract comma separated data from each data class. Updates these methods to utilize the Student class.

Functions:

- Needed to add student_data parameter to read_data_from_file(file_name: str, student_data: list):.
- Noticed in the starter file IO.input_student_data method, there were raise ValueErrors that have been handled in the Student class. So referencing the Lab03 code's IO.input_student_data, I simplified the code.

After testing the menu options and some error handling in PyCharm and the terminal, I found the program running as expected.

Conclusion

Module 07 presented how programmers can organize and manage data in Python. We have been working with lists of dictionary objects and have elevated that technique by creating a list of Student objects to gain more control of our data. Organizing the data into distinct classes makes it easier for other developers. While the user interaction with the program has not changed much over the past few modules, the methods we learned have developed the flow of our programming.