Efren Enriquez

08/14/2024

IT FDN 110A

Assignment 07

https://github.com/jefrene10/IntroToProg-Python-Mod07

Assignment 07 – Classes and Objects

Introduction

This week in IT FDN 110A the topic of functions was introduced. In this paper there will be a brief overview about some of the concepts learned during the module. Then the process used to write the script for this assignment will be further elaborated on throughout the paper. The content this week focused on classes and objects. This will include the use of statements, functions, classes, data class components such as attributes, constructors, and self-keyword, data validation, and inherited code. Finally discussing the initial code setup including setting a menu constant and defining variables, main body code, and testing of the code.

Concepts Learned

This week new concepts were explored into the core concepts of programming and version control. This includes the use of statements, functions, classes, data class components such as attributes, constructors, and self-keyword, data validation, and inherited code.

Data Class Components

This week we dove deeper into data classes and their components. Data classes can be used for example to process data to and from a data file. The data class typically has many components such as attributes, constructors, and properties, in addition to methods. Attributes in programming is a piece of data that is associated with an object. Object data (integers, strings, or custom data types) can be described or stored through the use of an attribute. Constructors or initializer is a special method that is automatically called when an object of a class is created with the main purpose of setting an object's value when it is created. Constructors are used in object oriented programming (OOP). The self keyword has been used in previous labs but never elaborated on until this module. The self keyword is included within the constructor method and it refers to data or functions found within an object instance and not directly in the class.

Data Validation

Any data can be placed into attributes dur to lack of any error handling. Private attributes can be used to kind of lock in attributes to not be changed outside of the class. To make an attribute private, two underscores were added before the attribute's name. Properties are functions that are designed to manage attribute data. Two properties are typically made, one for "getting" data and one for "setting" data, known as "Getters" and "Setters", also known as "Accessors" and "Mutators" respectively. To

indicate the “getter” the @property decorator is used. The “setter” decorator is .setter and this property function allows for the addition of validation and error handling.

Writing the Program

This weeks assignment built upon Assignment 06 but added new concepts learned such as advanced data collections such as statements, functions, classes, data class components such as attributes, constructors, and self-keyword, data validation, and inherited code. To start the Assignment06.py and Assignment07-Starter.py file was utilized as a reference and to practice using others code.

Constants and Variables

First the program constants were defined and were not changed throughout the program. Two constants were defined, MENU and FILE_NAME. MENU was a string that displayed the menu for a Course Registration Program and four selections: 1. Register a Student for a Course, 2. Show current data, 3. Save data to a file, and 4. Exit the program. FILE_NAME was created and set to a value of “Enrollments.json” to aid in creating the file with a correct name.

Next two global variables were created for this program. These variables include: students and menu_choice. There were other local variables that were utilized within the different functions created..

Classes and Functions

Various classes, properties, methods, and functions. This portion of the program was created incorporating new and old learning. First the Person class was created to represent the person data setting up properties for first and last name. Within this class, the properties to initialize the person object (__init__), getting the first and last name (first_name and last_name) with their respective “Getters” and “Setters”. Next the student class was created to represent the student data. Within this the properties for course name was made. The class file processor was similar to previous assignments, but was updated with new learning. The file processor class had various methods and functions such as the read_data_from_file, write_data_to_file. Next the IO class included the methods and functions such as output_error_messages, output_menu, input_menu_choice, output_student_courses, and input_student_data.

Main Body

The main body of the code is now much cleaner and organized with the use of the various class, variables, and methods created in the earlier part of the code. It was important to correctly make the intial code work so that the main goals were met this week.

Testing the Program

To successfully test the program, the program needed to do various things successfully. First was to take the users input for the first, last, and course name and then displaying it correctly (Choice 1 and 2). Then once done, saving the data to the .json file. Finally it is important that the code record, display, and save multiple registrations. The different error handling code integrated into the program was tested as a numerical value was inputted for the first and last name, no file being present to read, and errors while saving to the file. This was done successfully through executing the code both in PyCharm and the Command Shell.

Summary

In conclusion, the necessary tasks for Assignment 07 were completed. Tasks 1 through 3 were done to aid in learning and understanding the material for the week. Task 4 was completed through creating the script Assignment07.py. Finally task 5 was completed by writing this paper. Task 6 and 7 will be done following the completion of the documentation through posting files on GitHub. This week it was fun to learn more and build upon previous topics. This week the there was lots of content to learn and built upon previous knowledge at the same time. The program that was made this week was challenging but fun to make. To create the program this week the starter file was utilized and revised as this was encouraged. Learning more about statements, functions, classes, data class components, and various data validation. It was important to comment along the way as this was my feedback from the previous week. I was able to successfully implement new learning in achieving the correct outputs for this week's assignments.