Spencer Neiffer

ITFDN110A

3/11/2025

Assignment 07

https://github.com/SBNEIFFER/IntroToProg-Python-Mod07.git

Classes and Objects

Introduction

In this document, I will outline the steps I took to convert an existing Python program to use classes, objects, and properties to perform tasks. Much of the code for this script already existed, so I added code and edited existing portions. The result is a script that uses classes and objects to accomplish tasks.

Person Class

The first step I took in converting this script to classes and objects, was declaring a class called ***Person***. I also added a constructor method and gave it the string attributes of ***first_name***, ***last_name***, and ***course_name***. I then added ***first_name*** and ***last_name*** properties to the constructor and used double underscores to indicate that they're private attributes.

Next, I added a getter for the ***first_name*** property that returns the ***first_name*** with the first character capitalized. After this, I added a setter for the first_name that checks if the string value contains only letters. If it does or is blank, the ***first_name*** value is set, but a ***ValueError*** is raised if the string contains any numbers. I also a getter and setter for the ***last_name*** property using the same conditions.

Finally, I used the __str__ method to return a string containing the private ***first_name***, and ***last_name***, attributes from the ***Person*** class.

Inherited Student Class

I created a class called Student that inherits attributes from the original Person class. I then called the constructor and passed it the ***first_name***, ***last_name***, and ***course_name*** attributes using the ***super*** method. Then, I assigned the private ***course_name*** property using the ***course_name*** parameter.

My next step was to add a getter and setter for the ***course_name*** attribute. The getter returns the private ***course_name*** attribute with the first character capitalized, and the setter sets it to a string value.

I then used the __str__ method to return the ***first_name***, ***last_name***, and ***course_name*** attributes from the ***Student*** class.

Reading From JSON File

The next step I took was changing where the JSON file data loads. I switched in from the ***student_data*** dictionary to the list of dictionary objects called ***list_of_dictionary_data***. Then I used a ***For Loop*** to create student objects called ***student_object*** from the loaded JSON data and assigned them the ***Student*** attributes. Finaly, I appended the ***student_object*** dictionaries to the ***student_data*** list. This conversion was done because JSON doesn’t work with objects directly.

Writing to JSON File

In this function, I passed the student objects in ***student_data*** list into a dictionary called ***student_json*** using a For Loop. I then appended the ***student_json*** dictionaries to a list called ***list_of_dictionary_data*** and used the ***dump*** function to write it to the JSON file.

Displaying Current Data

When displaying the current data to the user, I changed print function from showing an F string containing student dictionaries, to a string that shows the student objects from the list ***student_data*** using a ***For Loop***.

Capturing User Input

The next step was to pass the captured user input to a student object called ***student*** instead of a dictionary. I assigned the ***Student*** class attributes: ***first_name***, ***last_name***, and ***course_name***, to the variables that hold the user input, then I appended the ***student*** object to the ***student_data*** list.

Summary

In conclusion, much of the code contained in this script was pre-existing. However, I converted it to use classes and objects instead of dictionaries. I created a Person class and a Student class that inherits attributes from the Person class. I then created constructors for each class and changed the dictionaries to objects. Whenever JSON data was read or written, I had to convert the objects back into dictionaries to work with the JSON. That was the extent of this code conversion.