Assignment07

Norm De Asis

Foundations Of Programming: Python

University of Washington

Instructor: Randal Root

March 12, 2025 (due)

March 15, 2025

Purpose

To create a Python program that demonstrates using constants, variables, and print statements to display a message about a student's registration for a Python course. This program is very similar to Assignment06, **but It adds set of data classes.**

Steps/Guide per Assignment Criteria

1. Definition of (Script) Header - this is given using the Assignment 07 starter Python file

- A script header with: Python 100 class, title: Assignment 07, assignment is to create a Python file using data classes and with structured error handling, change log, and author details.

2. Definition of Constants

- MENU as str
- FILE_NAME (as Constants) “Enrollments.json”

3. Definition of Variables

- Defined **students -** empty list and table of student data
- Defined **menu_choice** - empty string and hold the choice made by the user

4. Create Classes

4.1 Person Class

- Defined **first_name** and **last_name** as properties
- Instance:

person1 = Person()
person1.first_name = "Norm"
person1.last_name = "DA"

print(person1.first_name, person1.last_name, person1)

- Implemented getter and setter methods with simple validation
- Overrode __str__() to return formatted person data

4.2 Student Class

- Inherited from **Person** class
- Added **course_name** property with getter and setter.
- Overrode __str__() to return student data including the course name.

5. Create FileProcessor Class

- Applied static methods to handle file operations
- read_data_from_file(file_name: str): Reads student data from JSON file
- write_data_to_file(file_name: str, student_data: list): Writes student data to JSON file

- Added error handling for file operations (this is given per assignment criteria)

6. Create IO (Input/Output) Class – this displays applicable error and/or data

- output_error_messages:
- output_menu(menu: str):
- input_menu_choice():
- output_student_and_course_names(student_data: list):
- input_student_data(student_data: list):

7. Main Execution Logic

- Read student data from the JSON file at program start (choice)
- Display the menu in a loop until the user selects exit (to exit/stop = 4)
- Handle user input and execute respective functions based on menu choices:
    - **Option 1:** Register a student
    - **Option 2:** Display current student data
    - **Option 3:** Save student data to a file
    - **Option 4:** Exit the program

8. Implement Error Handling

- Added structured error handling in file operations, user input, and data processing to avoid runtime errors

9. Testing and Debugging

- Verified all functionalities by running different test cases
- Ensured the program correctly reads, writes, and displays student data
- Here are few (syntax, attribute…) errors I encountered:
    - Missing import json:
        - The script uses json.load(file) but json is not imported
        - Fix by adding import json at the beginning
    - @staticmethod indentation error
        - Fix by removing the indentation for @staticmethod
    - **student_data** is re-assigned but not returned properly:
        - The function takes **student_data** as an argument but reassigns it inside the function
        - This will not modify the original list outside the function
        - Fix was, either remove **student_data** from the parameter list or return the loaded data explicitly (chosen the latter)
    - Possible issue with **file.close()** in finally block:
        - file may not be defined if an error occurs while opening the file
        - Fix was, use a with **open(file_name, "r")** as file: statement to handle automatic closing
    - Duplicate while True loops:
        - The script contains four identical loops. Only one loop is needed
        - Fix was, duplicates must be removed
    - Inconsistent **menu_choice** Handling:

        - menu_choice = str(IO.input_menu_choice()) is used, make sure it's a string
        - In other places, menu_choice = IO.input_menu_choice() is used without conversion, which may cause errors if the function returns an integer
        - Fix: **always** convert **menu_choice** to a string, *menu_choice = str = ""*
    - Incorrect Menu Option Message in Some Places:
        - The message "Please only choose option 1, 2, or 3" does not include "4", a valid option in MENU
        - Fix was, I included "4"
        - The message "Please only choose option 1, 2, or 3" does not include "4", which is a valid option.
        - Fix: Ensure the message includes "4".
    - FileProcessor.read_data_from_file() signature mismatch:
        - Based on the script/code, read_data_from_file(file_name) only takes one argument
        - However, it is called with student_data=students, which might not be valid
        - Fix was to remove the student_data argument
    - Students is used before being defined:
        - students is/are assigned using FileProcessor.read_data_from_file(...), but students is/are passed as an argument before being initialized
        - Fix was, ensure students is/are defined before the line, or modify the function signature of read_data_from_file to avoid requiring student_data
    - Attribute error for Python Script: students = FileProcessor.read_data_from_file(FILE_NAME)
        - AttributeError: type object 'FileProcessor' has no attribute 'read_data_from_file'
        - Fix function call (removed file_name)

Note: Finding the solution and/or fix were difficult however, using different tools (PyCharm debugger, IDLE, AI, and Google) guided me to (in my opinion) fixing the errors.

References:

Class Module 07 materials.