Introduction to Programing with Python

Module 07 – Classes and Objects

Overview

In this assignment, you will learn about how to create and use classes to manage data. Course assignments help you learn through reading, watching demonstrations, performing programming in Python, and reflecting on what you learned through writing.

This assignment includes the following tasks:

- Read module text.
- Watch the module videos.
- Create a program.
- Document your knowledge.
- Submit your work.

Tip: Consider the following questions while you work through the module to help you focus:

- What are the differences between statements, functions, and classes?
- What is the difference between a data class, a presentation class, and a processing class?
- What is a constructor?
- What is an attribute?
- What is a property?
- What is class inheritance?
- What is an overridden method?
- What is the difference between Git and GitHub Desktop?

Task 1: Read the Module Notes and Watch Videos.

Start the assignment by reading the module's Notes document and watching the module's demonstration videos. You will find the Notes document within the Module07.zip file on the Canvas Module page. You will also find the links to videos in that same section.

Task 2: Watch the assignment videos.

Please watch the following video, in addition to the videos and demonstrations you watched in the module.

- Python OOP Tutorial 1: Classes and Instances (external site)
- PyCharm Version Control w/Git and GitHub (external site)
- 3 Simple Ways ChatGPT Can Make You a Better Coder (external site)

Task 3: Read about module topics

Please read the following articles, in addition to the text you read in the module.

1. python classes and objects (external site)

Task 4: Create a program

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.**

Note: Start by opening and reviewing the starter file Assignment07-Starter.py!

Acceptance Criteria

Your program must include the following features and code to be accepted as complete:

File Name:

- The file is named Assignment07.py

Script Header:

- The script header includes this text and has been updated with your name and the current date.

Constants:

- The constant **MENU: str** is set to the value:

```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course
    2. Show current data
    3. Save data to a file
    4. Exit the program
----------------------------------------
```

- The constant **FILE_NAME: str** is set to the value "Enrollments.json"
- Constant values do not change throughout the program.

Variables:

- **menu_choice: str** is set to empty string.
- **students: list** : list is set to and empty list

Classes:

- The program includes a class named FileProcessor.
- The program includes a class named IO.
- The program includes a class named Person.
- The program includes a class named Student.
- All classes include descriptive document strings.

Class Properties:

- The program includes properties for **student_first_name: str** and defaults to an empty string.
- The program includes properties for **student_last_name: str** and defaults to an empty string.
- The program includes properties for **course_name: str** and defaults to an empty string.
- The program's properties must include simple validation code.

Class Methods:

- The program includes a method to extract comma separately data from each data class.

Functions:

- All functions include descriptive document strings.
- All functions the include exception blocks use the output_error_messages() function for handling error messages.
- All non-instance functions use the @staticmethod decorator (The ones in the FileProcessor and IO classes.)
- The program includes functions with the following names and parameters:
    - output_error_messages(message: str, error: Exception = None)
    - output_menu(menu: str)
    - input_menu_choice()
    - output_student_courses(student_data: list)
    - input_student_data(student_data: list)
    - read_data_from_file(file_name: str, student_data: list):
    - write_data_to_file(file_name: str, student_data: list):

Input / Output:

- On menu choice 1, the program prompts the user to enter the student's first name and last name, followed by the course name, using the input() function and stores the inputs in the respective variables.
- Data collected for menu choice 1 is added to the **students** two-dimensional list of Student objects.
- On menu choice 2, the program uses the print() function to show a string of comma-separated values for each row collected in the **students** variable.

Processing

- When the program starts, the contents of the "Enrollments.json" are automatically read into a two-dimensional list of dictionaries rows using the json.load() function. Next, it converts that data into a list of Student object rows. (**Tip:** Make sure to put some starting data into the file or you will get an error!)
- On menu choice 3, the program opens a file named "Enrollments.json" in write mode using the open() function. Next, it converts the data in the **students** variable (a list of Student object rows) into a list of dictionary rows, then writes the list of dictionary data into the file using the json.dump() function. Finally, it closes the file using the close() method.
- On menu choice 4, the program ends.

Error Handling

- The program provides structured error handling when the file is read into the list of dictionary rows.
- The program provides structured error handling when the user enters a first name.
- The program provides structured error handling when the user enters a last name.
- The program provides structured error handling when the dictionary rows are written to the file.

Test:

- The program takes the user's input for a student's first, last name, and course name.
- The program displays the user's input for a student's first, last name, and course name.
- The program saves the user's input for a student's first, last name, and course name to a JSON file. (check this in PyCharm or a simple text editor like Notepad or TextEdit.)
- The program allows users to enter multiple registrations (first name, last name, course name).
- The program allows users to display multiple registrations (first name, last name, course name).
- The program allows users to save multiple registrations to a file (first name, last name, course name).
- The program runs correctly in both **PyCharm and** from the **console or terminal**.

Source Control:

- The script file and the knowledge document are hosted on a GitHub repository.
- A link to the repository is included in the knowledge document.
- A link to the repository is included in the GitHub links forum.

NOTE: The process and code needed to complete this assignment task is very similar to Modul07-Lab03!

Task 5: Document your knowledge

After you have created and tested your Python program, create a document **describing the steps you took in performing this assignment**.

- All resources for this assignment are found in the lectures, recommended reading, or recordings specified in the class syllabus. You do not need to locate additional resources outside of these.
- **Your document must conform to my professional document template to get full points!**
- Please save or download your document as a PDF file called Assignment07_YourNameHere.pdf.

Important: **Watch this video** to help you understand what I am looking for: Writing Professional Documents as needed.

Task 6: Post your Files to GitHub

In this module, you need to **post** your files on a public **GitHub repository** so that others may review it. Please post **both your knowledge document and your Python script file**.

Once, you understand how it works, perform the following to **create** a repository for your code:

a. **Login** to https://github.com (*Make a new account if needed!*)

b. **Create a new** repository called "***IntroToProg-Python-Mod07***".

c. **Upload** both of your files to the repository.

d. **Commit** the changes to save your work.

Task 7: Post a Link to GitHub

You will share your work using the Canvas discussion board. To do so, you must create a post with a link to your GitHub site. Other students will use this link to perform a peer review.

Important: Post only on the special discussion board called "Assignment 07 Documents for Review!"Please copy and paste the URL for your new GitHub site into your knowledge document. This makes grading a lot easier and is a big help! Thanks!

<Name>
<Date>
<Course>
<Assignment>
<GitHubURL>

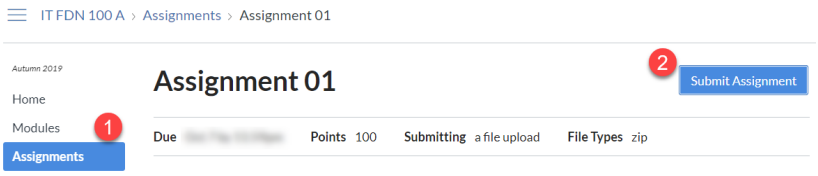
Title

Introduction

Lorem ipsum is a made-up language. It is used as a placeholder for a

Task 8: Submit your work

Now place your document with the Python script into a folder named A07, then compress the folder into a ".zip" file, before finally uploading the file to the class assignment page on Canvas.



Notes:

- Use the discussion board to request help on the assignment.
- The assignment can be completed using the lectures, assignment videos and reading, and module labs. You do not need to locate additional resources outside of the course material to complete it.
- Please read this article if you are unsure how to zip a folder How to Make a Zip File (external link)
- See the "**General Information and Helpful Tips**" module in Canvas for more help!

Step 9 - Perform a Peer Review (Not Graded!)

After you have posted your link to GitHub and submitted your assignment, go to the "Assignment 07 Documents for Review!" discussion board and **select another student's post and review.** Follow the link they posted and review their files on GitHub. **This is an informal review** that **does not affect** either your or their **grade**. **Try to pick someone's link that has NOT been reviewed yet, even if you have to wait a few days for one to appear!**

Notes:

- **Post** your comments as a reply to their posting so the review will be nested under the other student's posting.
- **Make sure** to say two things that you liked about their work
- **Make sure** to say one thing that could make the work better

Congratulations! You are done!