

Assignment 7

Contents

Introduction.....	1
Getting Started	1
Classes and Objects	2
First Draft of Assignment 7 Script.....	3
Attended Office Hours.....	4
Conclusions.....	5
References.....	5

Introduction

For our Assignment 7, we build on our first use of classes with more techniques. We are now fully into the tools of Object-Oriented Programming (OOP) with the creation of objects a major focus of this exercise. This assignment also includes the use of constructors, special methods, and use of the JSON library. The programming assignment continues to build on our work accomplished over the previous weeks. We will read a list of objects, instead of a list of dictionaries. Assignment 7 feels like we are becoming actual programmers.

Getting Started

The goals of this week's efforts are:

- Continuing the learning cycle, this time focusing on classes and objects.
- Start this document,
- Review the assignment,
- Draft the bones of the assignment in Pycharm,
- Continue using tools I have already learned.

Here's the Assignment 7 objective copied from the Module 7 text:

"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 Assignment

05, but It adds the use of functions, classes, and using the Separation of Concerns Pattern. Note: Start by opening and reviewing the starter file Assignment07-Starter.py!”

First impressions of assignment: We are making objects! Here’s some ideas:

- The script does not do anything new; we are just writing the instructions in a new way that will gain some benefits (which are indeterminate at this moment).
- Last week we added two classes, and we will add two more classes: Person and Student.
- I assume the new classes are the object classes.
- Because we are making objects. we will not be making a list of dictionaries, but a list of objects(?)
- The IO and FileProcessor will not have objects?

Classes and Objects

In reading about Classes and Objects in Dawson¹, there were a few points I see worth restating:

- A **class** is a design for an object (and while **class** is not capitalized, the name of the class is CAPITALIZED).
- An object combines characteristics (attributes) and behaviors (methods). Socratica’s video on classes likened the pattern to nouns (object), adjectives (attributes), and verbs (methods) – I’m ok with that!
- One communicates with Objects through method-parameters and return values.
- A unique object in a **class** is call an “instance”. Vis a vis, we say, “An instance of the Class”.
- *Instantiating* an object is easy, just follow this pattern:

```
variable_name = Classname(arguments)
```

```
i.e., critter_one = Critter(arguments)
```

I liked this summary from *Geeks for Geeks*:

“Classes and objects provide a way to model real-world entities and abstract concepts in code. They promote code **organization, encapsulation, inheritance, and polymorphism**, making complex systems easier to manage and extend.”

Notes from our class mentioned also the following:

- **Isolation:** Each object is isolated from others. Changes made to one object's data do not affect the data of other objects of the same class. This isolation helps maintain data integrity.
- **Reusability:** Classes serve as templates, and you can create as many objects as you need based on the same template. This reusability simplifies code development and maintenance.
- **Abstraction:** You can interact with objects based on their interfaces (the attributes and methods they expose) without worrying about how data is stored or processed internally. This abstraction simplifies the complexity of your code.

I’m seeing a lot of power in this tool. I have made a realization that the topic of Classes and Objects is probably too large to fully understand after a week.

¹ I am so very glad (!!!) that I had this book to read through on Wednesday, when we had no power, no internet, and the local batteries were going flat.

Based on the above learning, some further observations regarding Assignment 7:

- Our existing dictionary keywords (in Assignments 5 and 6) are now attributes assigned to the objects in a class.
- Many of Assignment 6's functions are now methods assigned to a class.
- I see we have four classes now – so four sets of instructions are available to create four kinds of objects?

Let's guess what they are (**Table 1**). Note that my initial thought was that only Students would be objects, but later revised that thought to include Students and Person. And actually, turns out they are all objects.

Table 1. Organization of classes and details for each to help with script.

Class Name (As assigned)	Description	Object Name	Variables and Parameters	Method
IO	Asks for input and handles errors in input or output.	IO	All_students_list	Output_error_messages() Input_menu_choice() Input_student_data() Welcome_str_menu()
FileProcessor	Moves data to and from a json file with json.dump json.load	FileProcessor	Json.file Note handling formatting here to read in the correct way.	Read_data_from_file() Write_data_to_file() Load_student_data
Person	Creates a Person Object	Person	first_name last_name	__init__() get_first_name() set_first_name() __str__()
Student	Inherits from the Person class and creates a Student Object. Adds course name.	Student	course_name _course_name first_name last_name	__init__() __str__() super().__init__() course_name()

Initially, I was unable to complete the above table...so there were some pieces missing in my understanding. More investigation was needed! Ultimately, I completed the table to my satisfaction.

First Draft of Assignment 7 Script

Jumping into the deep end! This time I'm going to look at the starter file to help understand more clearly understand the objective here. I see in the starter file that the instruction is that the Student class inherits from the Person Class, thus I will rearrange information in Table 1 above to reflect that. After adding Student Class and Person Class there are 51 red points in Pycharm and here's one more! Figure 1 shows how the script for Class Person looks at this point.

Reading *Real Python* tutorial on setter, getter, and properties, and have now I added these. Now there are only 13 red points in Pycharm and along with these I see several unresolved references, which I understand (generally) how to fix.

The big issue is the reading and writing to Json file.

I added new functions that separated the formatting of the data from the actual reading and writing chores. This helped with debugging because now I could see what was not working. Ah Ha! In one place I have a list format and at another location a dictionary format, which was causing issues. Other issues on tabs/spacing.

Figure 1. Class Person first draft. This does not work.

```
class Person:
    first_name: str
    last_name: str
    """ Base Class - Establishes a class of person objects with attributes of first and
    last name.

    Properties:
    -first_name(str): the person's first name.
    -last_name(str): the person's last name.

    """
    def __init__(self, first_name: str, last_name: str ):
        self._first_name = first_name
        self._last_name = last_name

    def get_first_name(self):
        return self._first_name.title()

    def set_first_name(self, value:str):
        if value.isalpha() or value == "":
            self._first_name = value
        else:
            raise ValueError("The first name should not be numerical. ")
```

Good things here: First effort here using get and set. While this is not quite correct, it's a good first try!

Other issues on tabs/spacing. And I'm going to use @property and @setter because it seems to work better.

I am reducing the red points slowly. Still having issues with understanding the JSON bit. Spend more time with additional resources. Found a Python tutorial video with Muppets! I'm in heaven!

Looks like the JSON formatting issues is called "serializing", more or less, and it gets complicated sometimes – much more than what I have here.

I've removed the static decorators from the Student and Person Classes – even though the assignment instructions say, "all functions" have this, that decorator is not helping. Turns out there are three general kinds of methods and Static is just one of those. Static is because it does not return anything – but I need to return data so Static doesn't work here.

Attended Office Hours

At office hours, what we are trying to accomplish in regard to saving data to JSON file was made clear. I can drop all the string formatting! String formatting is not needed for the file saving, even though the instructions mention that it is needed.

Now the script is looking much closer to what will work. No actual red points now in Pycharm.

Ok – NOW(!) it runs!! With only one warning. I had the same warning last week and just chose to ignore. I will ignore it again as time is running out.

Table 2. gives a Quality Assurance status.

Table 2. Status toward completing the assignment – Quality Assurance.

1. Read module text	Done
2. Watch the module videos.	Done. And supplemented with Dawson and Socratica, CS50 w/Muppets, Real Python and others.
3. Create a program	Program created and it does work!
4. Document your knowledge and make a PDF.	Make a PDF
5. Submit your work to discussion board.	Submit just the URL GitHub - Done
6. Post Files to GitHub	Done
7. Post Link to GitHub Files on Canvas	Done
8. Post per usual to Canvas	Done
9. Make a Peer Review (optional)	

Conclusions

Assignment 07 focused on producing a script that incorporates Objects, and use of JSON files for saving and reading. Importantly, this assignment had our first interface with many of the tools for objects. As a side effect of working so much with the script this week, I am furthering my understanding of Pycharm and using more of the tools offered.

This was a very challenging assignment due to the large number of new concepts to understand and apply. I was able to gain a preliminary understanding of objects. I can see there is much more to learn about Objects in Python.

References

Dawson, Michael. 2010. Python Programing for the Absolute Beginner. Third Edition. Published by Course Technology PTR a part of Cengage Learning. Boston, MA, USA.