

Michael Okimura

05/17/2024

Foundations of Programming - Python

Assignment 06

<https://github.com/MOkimura/IntroToProg-Python-Mod06>

# Separating Functions Inside Classes with “Jason”

## Introduction

This was the most challenging assignment yet due to the difficulty of understanding how classes work but more so what role do they play in coding. I read many articles and watched many videos on Python classes and classes in general. I was unable to find a source that was able to explain it in a manner that I could understand easily. I had to take small bits of information for each source to piece together everything so I could complete this assignment.

## Creating the Program

### Program Objective

This week, we needed to utilize functions inside 2 different classes to execute the same processes as we have been so far. This dramatically changed the code along with incorporating json files for reading and writing the data. Lastly, error handling was included as well.

### Input / Output

All these functions were called from inside the input / output (IO) class. Once each function was broken down into their individual pieces, it was just like writing the code as before. The inputs were the same as we did at the beginning of the course, print statements, etc. Same with adding error handling as we did in the last assignment. Once the functions were completed, it was just about placing them into the menu options by calling them with the class.

### Processing

The file, as before, was auto read into the program upon starting along with providing an error message if no file existed but the program would still run. Writing the data to the json file was successful as well. The bits of code that are still a bit unknown to me are the double underscore items. The best explanation I can provide after researching them is they are preset variables versus private variables, meaning they will store data from locations that are predetermined versus having us tell it specifically what to store.

```
111 # Start of program
112 if __name__ == "__main__":
113     students = FileProcessor.read_data_from_file(file_name=FILE_NAME, student_data=students)
```

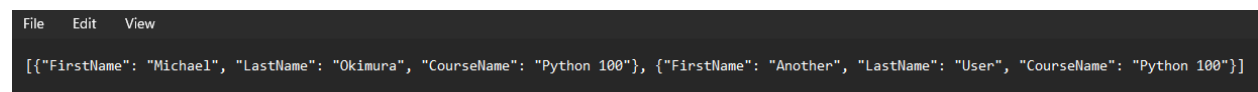
Figure 1: Program start using double underscore code.

## Error Handling

There are 4 items required to address in this assignment. They are basically located in pairs. Inputting the first and last name but as before I included the course name with it. I told the code that if left blank, make the user enter a value in of any kind, it just can't be empty. For the data read and write, I put code in for if the file doesn't already exist to display an error however the program will still run. In the case of writing the data, the context manager was set to "write" mode so the error won't trigger if it doesn't exist. If set to "append" or "read" it would though.

## Testing

I tested the program using Windows in both Visual Studio Code and Windows Command Line. In comparison to last week's assignment, I got the same results from both. The multiple inputs, displaying of the data, and file reading/writing all worked correctly.

A screenshot of a code editor window with a dark background. The menu bar at the top shows 'File', 'Edit', and 'View'. The main text area contains a single line of JSON data: [{"FirstName": "Michael", "LastName": "Okimura", "CourseName": "Python 100"}, {"FirstName": "Another", "LastName": "User", "CourseName": "Python 100"}].

```
File Edit View  
[{"FirstName": "Michael", "LastName": "Okimura", "CourseName": "Python 100"}, {"FirstName": "Another", "LastName": "User", "CourseName": "Python 100"}]
```

**Figure 2: Sample data written to the json file using Windows Command Line.**

## Summary

The classes didn't add a lot more complexity to coding. Based on all the information I took in about them it is something that is generally difficult to explain and show how it works. Calling a function from inside a class, in my opinion, is not a difficult concept to understand. The challenge was to understand it in the big picture of the program. There were 2 classes with 7 different functions that were sort of bouncing around up and down the code which made it difficult to understand and follow. It was easier to write the code by focusing on writing 1 function at a time, then putting them into their respective classes, and then seeing where each needed to go. Working at it piece by piece helped reduce any frustration I had in trying to learn how it all worked together. And quick, last note, I have been trying to shorten my write-ups since I've been verbose in the previous ones and I feel like I achieved that here however I hope it wasn't too short and not explicative enough with regards to explaining what I did to satisfy the requirements of this assignment.