Nathan Castek

February 20th, 2024

IT FDN 110 A

Assignment06

https://github.com/trinathlon/IntroToProg-Python-Mod06

Functions

Introduction

In this assignment, I reviewed various articles and videos detailing the usage of classes, functions, working with JSON files, how to handle errors, and how to use GitHub for configuration management. Then, I created a simple python script that prompts a user to make a selection from a menu of options. Depending on the selection, the program will execute the appropriate command. In this assignment, I utilized: constants, variables, conditional operators, the input function, the print function, type hints, while loops, lists, dictionaries, error handling, functions, classes and worked with JSON files. Furthermore, I practiced testing my code from the integrated development environment PyCharm and from the terminal.

Research

I started out the assignment by reading the various articles and watching the videos for this week's module and assignment. I learned about classes, functions, error handling, working with JSON files, and hot to utilize GitHub for configuration management.

Programming

In the next part of the assignment I created a simple python program that prompts a user to make a selection from a menu of options and then performs a specific action based on the selection the user made.

Header

I started out my program by revising the header file provided in the assignment starter file. I removed the professors name and replaced it with my name and date. The resultant header can be seen below.

```
# ------------------------------------------------------------------------
--------------- #
# Title: Assignment06
# Desc: This assignment demonstrates using functions
# with structured error handling
# Change Log: (Who, When, What)
#   NCastek,2/20/2024,Created Script
# ------------------------------------------------------------------------
--------------- #
```

Defining Imports, Constants, and Variables

I then defined the imports, constants, and variables for the program. In this assignment I knew I had to read and write to a JSON file. Therefore, I needed to import json in order to use the json functions. Next, I defined the constants and variables. The constants and variables were detailed in the assignment, so I ensured the names were identical to the assignment instructions. The resultant code can be seen below.

```
import json

# Define the Data Constants
MENU: str = '''
---- 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.
----------------------------------------
'''
# Define the Data Constants
FILE_NAME: str = "Enrollments.json"

# Define the Data Variables and constants
students: list = []  # a table of student data
menu_choice: str = ''  # Hold the choice made by the user.
```

Define Classes

After revising the header, defining the constants, and variables, I set up the code to define the FileProcessor and the IO class defined in the assignement instructions. I created each class, made a docstring describing the class, and then used the “pass” keyword as a placeholder.

Define Functions

After creating placeholder for the two classes, I began creating functions under each class. I was able to create functions by leveraging code from the starter file and from the module notes. The first function I defined was the read_data_from_file function in the FileProcessor class. I used a try/except/finally block to read the file, except any error messages, then close the file. Lastly, I returned student_data at the end of the function. The resultant function can be seen below:

```
def read_data_from_file(file_name: str, student_data: list):
    """ This function reads data from a file

    ChangeLog: (Who, When, What)
    NCastek,2/20/2024,Created function

    :return: list
    """
    try:
        file = open(file_name, "r")
        student_data = json.load(file)
        file.close()
    except Exception as e:
        IO.output_error_messages(message="There was a problem with reading
the file.", error=e)
    finally:
        if file.closed == False:
            file.close()
    return student_data
```

Next I created the write_data_to_file function. I followed the same method as before by trying to write to the file, excepting any errors, then closing the file. The resultant code can be seen below:

```
def write_data_to_file(file_name: str, student_data: list):
    """ This function writes data to a file

            ChangeLog: (Who, When, What)
            NCastek,2/20/2024,Created function

            :return: None
    """
```

```
    try:
        file = open(file_name, "w")
        json.dump(student_data, file)
        file.close()
        IO.output_student_courses(student_data=student_data)
    except Exception as e:
        IO.output_error_messages(message="There was a problem with writing
data to the file.", error=e)
    finally:
        if file.closed == False:
            file.close()
```

I then created the functions for the IO method. I started with the output_error_message function. The resultant code can be seen below:

```
@staticmethod
def output_error_messages(message: str, error: Exception = None):
    """ This function displays a custom error messages to the user

    ChangeLog: (Who, When, What)
    NCastek,2/20/2024,Created function

    :return: None
    """
    print(message, end="\n\n")
    if error is not None:
        print("-- Technical Error Message -- ")
        print(error, error.__doc__, type(error), sep='\n')
```

Next, output_menu printed out the menu:

```
@staticmethod
def output_menu(menu: str):
    """ This function displays the menu of choices to the user

    ChangeLog: (Who, When, What)
    NCastek,2/20/2024,Created function

    :return: None
    """
    print()  # Adding extra space to make it look nicer.
    print(menu)
    print()  # Adding extra space to make it look nicer.
```

Input_menu_choice was a function used to have the user select a menu choice:

```
@staticmethod
def input_menu_choice():
    """ This function gets a menu choice from the user

    ChangeLog: (Who, When, What)
    NCastek,2/20/2024,Created function

    :return: string with the users choice
```

```
    """
    choice = "0"
    try:
        choice = input("Enter your menu choice number: ")
        if choice not in ("1", "2", "3", "4"):  # Note these are strings
            raise Exception("Please, choose only 1, 2, 3, or 4")
    except Exception as e:
        IO.output_error_messages(e.__str__())  # Not passing e to avoid the
technical message

    return choice
```

Output_student_courses was used to loop through all the students and print out their information to the screen:

```
@staticmethod
def output_student_courses(student_data: list):
    """ This function displays the student courses to the user

    ChangeLog: (Who, When, What)
    NCastek,2/20/2024,Created function

    :return: None
    """
    # Process the data to create and display a custom message

    print("-" * 50)
    for student in student_data:
        print(f'Student {student["FirstName"]} '
              f'{student["LastName"]} is enrolled in
{student["CourseName"]}')
    print("-" * 50)
```

Input_student_data utilized the input function to gather information from the user and store the information and the respected variable names, that was then combined into an individual student. The resultant code can be seen below:

```
@staticmethod
def input_student_data(student_data: list):
    """ This function gets the first name, last name, and course name from
the user

    ChangeLog: (Who, When, What)
    NCastek,2/20/2024,Created function

    :return: list
    """

    try:
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError("The first name should only contain alphanumeric
characters.")
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
```

```
            raise ValueError("The last name should only contain alphabetical
characters.")
        course_name = input("Please enter the name of the course: ")
        student = {"FirstName": student_first_name,
                   "LastName": student_last_name,
                   "CourseName": course_name}
        student_data.append(student)
        print(f"You have registered {student_first_name} {student_last_name}
for {course_name}.")

    except ValueError as e:
        IO.output_error_messages(message="That value is not the correct type
of data!", error=e)
    except Exception as e:
        IO.output_error_messages(message="There was a non-specific error!",
error=e)
    return student_data
```

Read JSON File

After defining all the classes and functions, I set up the code to read in the Enrollments.json file. I did this by calling the read_data_from_file function I previously created. I used the arguments FILE_NAME and students and stored the returned value in the variable students. The resultant code can be seen below:

```
students = FileProcessor.read_data_from_file(file_name=FILE_NAME,
student_data=students)
```

Menu Selection

After reading in the JSON file, I set up the code to obtain user menu selection. I knew I always wanted the menu selection to pop up first until the user hit menu option ‘4’. Therefore, I decided to create a “while” loop that would always be “True” until the user selected menu option ‘4’.

I started out the menu selection with “while True:” and then indented under that loop I printed the constant “MENU” by calling the output_menu function. Next, I prompted the user for input using the input_menu_choice function and stored their input in the variable “menu_choice”. The resultant code can be seen below:

```
while True:

    # Present the menu of choices
    IO.output_menu(menu=MENU)
    menu_choice = IO.input_menu_choice()
```

If/Else Statements

After obtaining user's input for menu selection, I created a series of if/elif/else statements for all the options the user could have picked. My first block of code was: if menu_choice == "1". In this scenario I called the input_student_data function by passing in the argument students and returning the result in the variable students. The resultant code can be seen below:

```
# Input user data
if menu_choice == "1":
    students = IO.input_student_data(student_data=students)
    continue
```

My next block of code was for if the user selected menu option two. I utilized an "elif" statement because I knew there were more options to select from, so did not want to use an else statement here. I then made a function call to the output_student_courses function by passing in the argument students and returning the result in the variable students. The resultant code can be seen below:

```
# Present the current data
elif menu_choice == "2":
    IO.output_student_courses(student_data=students)
    continue
```

My next block of code was for if the user selected menu option three. I utilized an "elif" statement because I knew there were more options to select from, so did not want to use an else statement here. Furthermore, I called the write_data_to_file function by passing in the argument FILE_NAME and students. The resultant code can be seen below:

```
# Save the data to a file
elif menu_choice == "3":
    FileProcessor.write_data_to_file(file_name=FILE_NAME,
student_data=students)
    continue
```

My next block of code was for if the user selected menu option four. I utilized an "elif" statement because I knew there were more options to select from, so did not want to use an else statement here. In this block I used the word "break" to break out of the "while" loop at the beginning of the program. The resultant code can be seen below:

```
elif menu_choice == "4":
    break  # out of the loop
```

My last block of code was used as an input error catching block. The user was supposed to input 1, 2, 3, or 4. However, there is a chance that they could have entered something else. Therefore, I created an “else” statement that would catch any user input that was not 1, 2, 3, or 4. In this block I simply used the print function to notify the user that they did not enter a valid option.

```
else:
    print("Please only choose option 1, 2, or 3")
```

Testing

After completing my program I tested it by running it from PyCharn and the terminal. I first ran my program from PyCharm and below you can see my user input’s and what was printed to the screen.

Note: I first made the entry:

```
[
{"FirstName": "Bob", "LastName": "Smith", "CourseName":  "Python 100"},
{"FirstName": "Sue", "LastName": "Jones", "CourseName":  "Python 100"}
]
```

in the Enrollments.json file so the program could read in the file.

Next, I tested my program in the terminal and saw identical results to when I used PyCharm.

```
---- 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.
----------------------------------------


Enter your menu choice number: 2
-------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
-------------------------------------------------


---- 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.
----------------------------------------


Enter your menu choice number: 1
Enter the student's first name: Daniel
Enter the student's last name: Lewis
Please enter the name of the course: Python 100
You have registered Daniel Lewis for Python 100.


---- 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.
----------------------------------------


Enter your menu choice number: 2
------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Daniel Lewis is enrolled in Python 100
------------------------------------------------


---- 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.
----------------------------------------


Enter your menu choice number: 3
------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Daniel Lewis is enrolled in Python 100
------------------------------------------------
```

```
---- 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.
----------------------------------------

Enter your menu choice number: 1
Enter the student's first name: Sarah
Enter the student's last name: True
Please enter the name of the course: Python 100
You have registered Sarah True for Python 100.

---- 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.
----------------------------------------

Enter your menu choice number: 2
------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Daniel Lewis is enrolled in Python 100
Student Sarah True is enrolled in Python 100
------------------------------------------------

---- 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.
----------------------------------------

Enter your menu choice number: 3
------------------------------------------------
Student Bob Smith is enrolled in Python 100
```

```
Student Sue Jones is enrolled in Python 100
Student Daniel Lewis is enrolled in Python 100
Student Sarah True is enrolled in Python 100
--------------------------------------------------


---- 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.
----------------------------------------


Enter your menu choice number: 1
Enter the student's first name: 5
That value is not the correct type of data!

-- Technical Error Message --
The first name should only contain alphanumeric characters.
Inappropriate argument value (of correct type).
<class 'ValueError'>


---- 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.
----------------------------------------


Enter your menu choice number: 1
Enter the student's first name: Ben
Enter the student's last name: 6
That value is not the correct type of data!

-- Technical Error Message --
The last name should only contain alphabetical characters.
Inappropriate argument value (of correct type).
<class 'ValueError'>


---- 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.
----------------------------------------


Enter your menu choice number: 4
Program Ended

Process finished with exit code 0
```

Summary

In this assignment, I reviewed various articles and videos detailing the usage of classes, functions, working with JSON files, how to handle errors, and how to use GitHub for configuration management. Then, I created a simple python script that prompts a user to make a selection from a menu of options. Depending on the selection, the program will execute the appropriate command. In this assignment, I utilized: constants, variables, conditional operators, the input function, the print function, type hints, while loops, lists, dictionaries, error handling, functions, classes and worked with JSON files. Furthermore, I practiced testing my code from the integrated development environment PyCharm and from the terminal.