

Assignment 06: Functions

Introduction

In this document I will cover the completion of assignment six for Foundations of Programming – Python, functions and classes. The follow topics will be discussed within this document.

- Terms to know
- Steps taken for assignment 06
- Summary of learnings

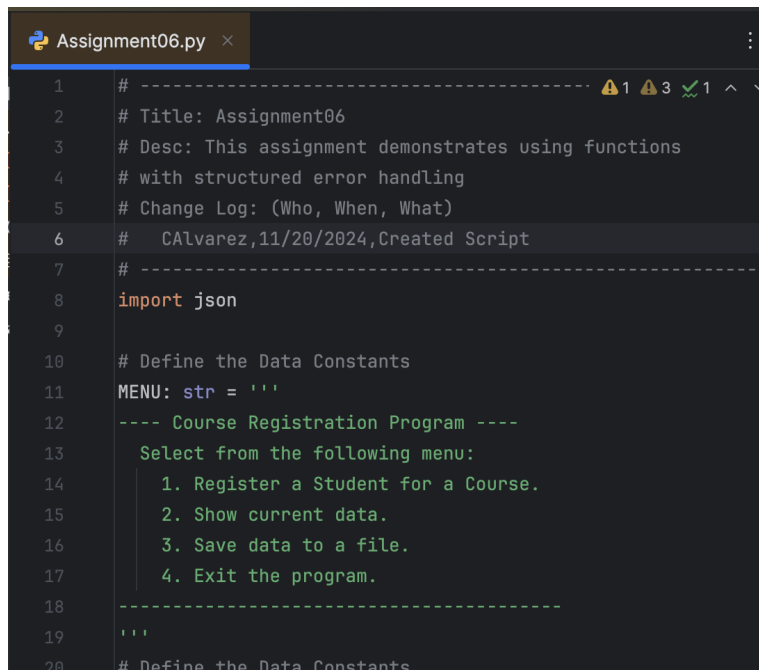
Terms to Know

Classes: in Python, classes are a form of grouping variables, functions and constants by way of their class. (cited from Mod06 notes)

Functions: a block of code that will run only when it is called upon. One can pass known parameters and data into a function (cited from W3schools.com)

Steps Taken for Assignment 06

For assignment six we were provided with starter code. I opened the starter code in PyCharm and updated the starter code header to reflect the work I would be completing. See figure 1.



```
1  # -----
2  # Title: Assignment06
3  # Desc: This assignment demonstrates using functions
4  # with structured error handling
5  # Change Log: (Who, When, What)
6  #   CALvarez,11/20/2024, Created Script
7  # -----
8  import json
9
10 # Define the Data Constants
11 MENU: str = '''
12 ---- Course Registration Program ----
13     Select from the following menu:
14     1. Register a Student for a Course.
15     2. Show current data.
16     3. Save data to a file.
17     4. Exit the program.
18     -----
19     '''
20 # Define the Data Constants
```

Figure 1. Started code updated

Next I added the two classes, File Processor and IO. Pictured below.

```
Assignment06.py x
31 file = None # Holds a reference to an opened file.
32 menu_choice: str # Hold the choice made by the user.
33
34 # Processing -----
35 class FileProcessor:
36     """
37     A collection of processing layer of functions that work with Json files
38
39     ChangeLog: (Who, When, What)
40     CAlvarez,1.1.2024,Created Class
41     """
42     pass
43
44 # Presentation -----
45 class IO:
46     """
47     A collection of presentation layer functions that manage user input and output.
48
49     ChangeLog: (Who, When, What)
50     CAlvarez,1.1.2024,Created Class
51     CAlvarez,1.1.2024,Added menu output and input functions
52     CAlvarez,1.1.2024,Added a function to display the data
53     CAlvarez,1.1.2024,Added a function to display custom error messages
54     """
```

Figure 2. Classes added to code

The function that was used in the labs for module six was also added to the code. See figure 3.

```
51
52
53 @staticmethod
54 def output_error_messages(message: str, error: Exception = None)
55     """ This function displays a custom error message to the user
56
57     ChangeLog: (Who, When, What)
58     CAlvarez,11.18.2024,Created Class
59
60     :return: None
61     """
62     print(message, end="\n\n")
63     if error is not None:
64         print("--Technical Error Message--")
65         print(error, error.__doc__, type(error), sep='\n')
66
```

Figure 3. Function added

From here I added the output menu and updated where the output menu would be used. Figure 4 and 5 below.

```

67     @staticmethod
68     def output_menu(menu: str):
69         """ This function displays the menu of choices to the user.
70
71         ChangeLog: (Who, When, What)
72         CAIvarez,11.18.2024, Created Class
73
74         :return: None
75         """
76         print() # Space added for aesthetics
77         print(menu)
78         print() # Space added for aesthetics
79

```

Figure 4. Added output menu

```

109     # Present and Process the data
110     while (True):
111
112         # Present the menu of choices
113         IO.output_menu(menu=MENU)
114

```

Figure 5. Updated usage of menu

Added the output menu function. See figure 6. And updated in the code where this function would be called. See figure 7.

```

80     @staticmethod
81     def input_menu_choice():
82         """ This function gets a menu choice from the user.
83
84         ChangeLog: (Who, When, What)
85         CAIvarez,11.18.2024, Created Class
86
87         :return: string with the users choice
88         """
89         choice = "0"
90         try:
91             choice = input("Enter your menu choice number: ")
92             if choice not in ("1", "2", "3", "4"):
93                 raise Exception("Please, choose only 1,2,3, or 4")
94         except Exception as e:
95             IO.output_error_messages(e.__str__()) # Not passing e to a
96
97         return choice
98

```

Figure 6. Output menu function

```

131
132     # Present the menu of choices
133     IO.output_menu(menu=MENU)
134
135     menu_choice = IO.input_menu_choice()
136

```

Figure 7. Output menu usage

From here I added the function to support the output of the student course and made the updates to support this function to menu choice two. See figure 8 and 9.

```

99     @staticmethod
100     def output_student_and_course_name(student_data: list):
101         """ This function displays the student and course name
102
103         ChangeLog: (Who, When, What)
104         CAlvarez,11.18.2024, Created Class
105
106         :return: None
107         """
108         print("-" * 50)
109         for student in student_data:
110             print(f'student {student["FirstName"]}'
111                   f'{student["LastName"]} is enrolled in {student["CourseName"]}')
112         print("-" * 50)
113

```

Figure 8. Function added for student course output

```

181     # Present the current data
182     elif menu_choice == "2":
183
184         # Updated to use function output_student_course
185         IO.output_student_and_course_name(students)
186         continue
187

```

Figure 9. Menu choice two updated to above function

The next function was for input of the student data. I recalled some of the work from one of the labs and carried over the learnings here. Figure 10 and 11 show the updates to support input of student data and the update to menu choice one.

```

114     @staticmethod
115     def input_student_data(student_data: list):
116         """ This function gets the first name, last name and a course name.
117
118         ChangeLog: (Who, When, What)
119         CAvarez,11.18.2024,Created Class
120
121         :return: list
122         """
123
124         try:
125             student_first_name = input("Enter the student's first name: ")
126             if not student_first_name.isalpha():
127                 raise ValueError("The first name should not contain numbers.")
128             student_last_name = input("Enter the student's last name: ")
129             if not student_last_name.isalpha():
130                 raise ValueError("The last name should not contain numbers.")
131             course_name = input("What is the student's course name? ")
132             student = {"FirstName": student_first_name,
133                       "LastName": student_last_name,
134                       "CourseName": course_name}
135             student_data.append(student)
136             print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
137         except ValueError as e:
138             IO.output_error_messages(message="Please check data entered one of the values was incorrect.", Exception=e)
139         except Exception as e:
140             IO.output_error_messages(message="Error: There was a problem with the data entered", Exception=e)

```

Figure 10. Input student data function

```

178
179     # Input user data
180     if menu_choice == "1": # This will not work if it is an integer!
181         students=IO.input_student_data(student_data=students)
182         continue
183

```

Figure 11. Menu choice one update

Next, I worked on the read data from file function. Figure 12 depicts the update to the file processing field.

```

35     class FileProcessor:
42     def read_data_from_file(file_name: str, student_data: list):
43         """
44         ChangeLog: (Who, When, What)
45         CAvarez,11.18.2024,Created function
46
47         :return: list
48         """
49
50         try:
51             file = open(file_name, "r")
52             student_data = json.load(file)
53             file.close()
54         except Exception as e:
55             IO.output_error_messages(message="Error: There was a problem with reading the file.", error=e)
56
57         finally:
58             if file.closed == False:
59                 file.close()

```

Figure 12. Function for read data file

I completed the last function for the assignment which was to write the data to file. I added the function to support this, see figure 13 and 14.

```

62     def write_data_to_file(file_name: str, student_data: list):
63         """ This function writes data to json file
64
65         ChangeLog: (Who, When, What)
66         CALvarez,11.18.2024, Created function
67
68         :return: none
69         """
70         try:
71             file = open(file_name, "w")
72             json.dump(student_data, file)
73             file.close()
74             IO.output_student_and_course_name(student_data=student_data)
75         except Exception as e:
76             message = "Error: There is a problem with writing to file.\n"
77             message += "Please check that the file is not open in another program."
78             IO.output_error_messages(message=message, error=e)
79
80         finally:
81             if file.closed == False:
82                 file.close()

```

Figure 13. Function to write to data file

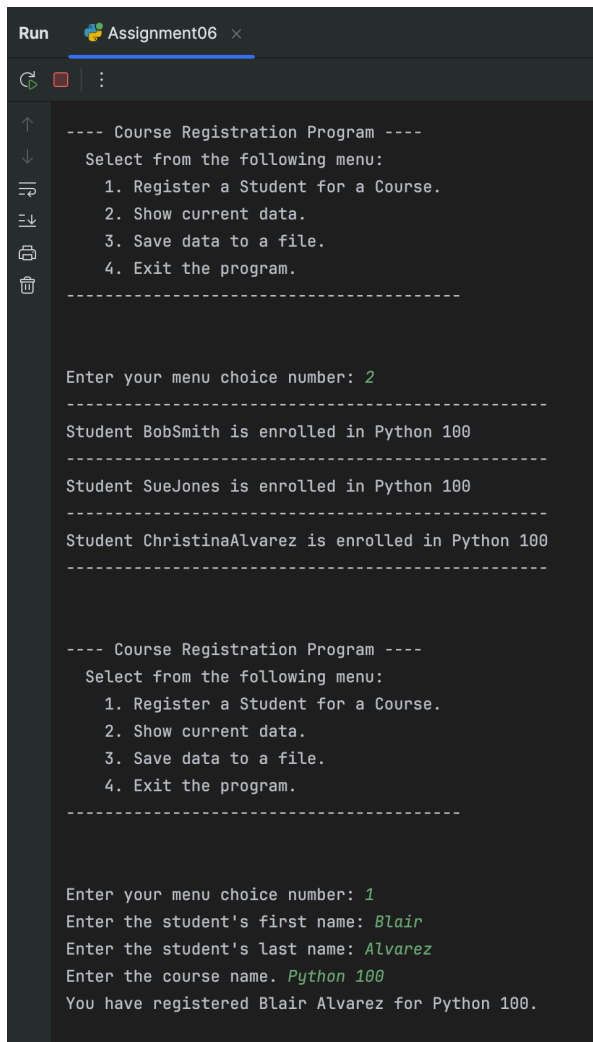
```

207     # Save the data to a file
208     elif menu_choice == "3":
209
210         FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
211         continue
212

```

Figure 14. Update to menu choice three

Lastly, I ran the code to ensure all was working as expected. See figure 15. And uploaded to GitHub, <https://github.com/Christina64/IntroToProg-Python-Mod06>.



```
Run Assignment06 x
---- 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 BobSmith is enrolled in Python 100
-----
Student SueJones is enrolled in Python 100
-----
Student ChristinaAlvarez 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: Blair
Enter the student's last name: Alvarez
Enter the course name: Python 100
You have registered Blair Alvarez for Python 100.
```

Figure 15. Check to ensure code runs

Summary of learnings

Assignment six was a more challenging assignment ensuring that functions were accurately represented and that the menu choices were updated along the way to eliminate duplication. Observations from this assignment is that functions can simplify the coding process by using code that was defined in previous projects. Additionally, functions seem to make code relatively top heavy and the actual body of the code a lot leaner. One big learning from this assignment was the importance of indentation.