

Jennifer Bailey

08/19/2025

Foundations of Programming: Python - Assignment 0

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

Creating a student registration tool using Python: Adding functions and classes and implementing Separation of Concerns

This assignment asked students to create a Python script that would prompt users to input student registration data (first and last names, course), allow the user to see the data they input, save the data to a .csv file, and exit the program when finished entering enrollments. In addition, students were asked to write functions and organize them into classes focused on, respectively, file processing and input/output.

Method

The full Python script is shown in Appendix 1. Goals of the assignment included understanding and constructing reusable functions to streamline code and organizing the script to maximize flexibility and clarity. The program begins by defining constants and variables. Next, functions related to file handling are created and assigned to the class **FileProcessor**. Functions related to obtaining and displaying user input were created and assigned to the class **IO**. Finally, the main body of the script calls the functions necessary to complete program test criteria.

Program test criteria were:

- 1) The program takes the user's input for a student's first, last name, and course name (option 1).
 - a. This was accomplished using a function (**output_menu**) that displays the menu to the user and a second function (**input_menu_choice**) that collects the desired user action. Finally, a third function (**input_student_data**) was called using `IO.input_student_data` to collect user input (student full name and course), collect the data into a dictionary called **student** and append the newly entered data to **student_data**.
 - b. Allowing multiple entries was achieved by using a while loop, which repeatedly presents the menu, calls the relevant functions, and appends new data entries to **student_data** until the user chooses to exit the program.
 - c. Error handling was implemented using a custom **output_error_messages** function that prints appropriate error messages when exceptions are encountered.
- 2) The program displays the user's input for a student's first, last name, and course name. Note: this was achieved in two ways.

- a. Each time new student data are entered using option 1, the **input_student_data** function outputs the information that was just entered: "You have registered {student_first_name} {student_last_name} for {course_name}."
 - b. Upon selection of option 2, the **output_student_data** function is called. This function displays a comma separated list of student names and courses, which includes the data imported from the starting JSON file as well as (all) new names and courses entered by the user.
- 3) The program saves the user's input for a student's first, last name, and course name to a JSON file.
 - a. The **write_data_to_file** function, which is assigned to the **FileProcessor** class, uses a for loop and the json.dump function to save user inputs to the file Enrollments.json. The for loop means that data from multiple student registrations can be saved.
- 4) The program allows users to enter multiple registrations (first name, last name, course name).
 - a. Please see 1b above.
- 5) The program allows users to display multiple registrations (first name, last name, course name).
 - a. Please see 2b above.
- 6) The program allows users to save multiple registrations to a file (first name, last name, course name).
 - a. Please see 3a above.
- 7) The program runs correctly in both **PyCharm** and from the **console or terminal**.
 - a. The script was run twice, once using the PyCharm integrated development environment and once from MacOS Terminal (see Results, Appendix 2, Appendix 3).

Finally, the current paper, the saved JSON file, and the script displayed in Appendix 1 were uploaded to GitHub here: <https://github.com/Jen-Bailey/IntroToProg-Python-Mod06> .

Results

The script worked as intended. Appendix 2 shows the output from PyCharm and Appendix 3 shows the output from MacOS Terminal.

Discussion

The current paper describes the steps necessary to complete the assigned task: to write a Python script that uses Separation of Concerns, classes, and functions to allow a user to enter, view, and save registration data in a JSON file for multiple students, including student first and last name and course. The included script (Appendix 1) successfully accomplishes the assigned task, as shown in the outputs in Appendix 2 and 3.

Appendix 1. Full Python script.

```
# ----- #
# Title: Assignment06
# Desc: This assignment demonstrates using functions
# with structured error handling
# Change Log: (Who, When, What)
#   RRoot,1/1/2030,Created Script
#   J Bailey,8/19/35,Edited Script
# ----- #
import json
import io

# 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.
-----
"""
FILE_NAME: str = "Enrollments.json"

# Define the Data Variables
#student_first_name: str = " # Holds the first name of a student entered by the user.
#student_last_name: str = " # Holds the last name of a student entered by the user.
#course_name: str = " # Holds the name of a course entered by the user.
#student_data: dict = {} # one row of student data
students: list = [] # a table of student data
#file = None # Holds a reference to an opened file.
menu_choice: str = " # Hold the choice made by the user.

# Processing -----#
class FileProcessor:
    """
    This class includes functions related to handling JSON files

    ChangeLog: (Who, When, What)
    J Bailey, 8/19/25, Created Class
    J Bailey, 8/19/25, Added functions to read and write JSON data
    """

    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        """ This function loads data from a JSON file

        ChangeLog: (Who, When, What)
        J Bailey, 8/19/25, Created function

        :return: list of student data retrieved from JSON file
        """
        try:
            file = open(file_name, "r")
```

```

        student_data = json.load(file)
        file.close()
    except FileNotFoundError as e:
        IO.output_error_messages(
            "Text file must exist before running this script!", e)
    except Exception as e:
        IO.output_error_messages(
            "There was a non-specific error!", e)
    finally:
        if file.closed == False:
            file.close()
    return student_data

```

```

@staticmethod
def write_data_to_file(file_name: str, student_data: list):
    """ This function writes data entered by user
    into a JSON file and displays the data that was saved

```

ChangeLog: (Who, When, What)
J Bailey, 8/19/25, Created function

```

: return: None
"""

```

```

try:
    file = open(file_name, "w")
    json.dump(student_data, file)
    print(f'Data written to {file_name}:')
    for student in student_data:
        print(student['FirstName'], student['LastName'],
              student['CourseName'])
    file.close()
except TypeError as e:
    IO.output_error_messages("Please check that the data is a valid JSON format", e)
except Exception as e:
    IO.output_error_messages("There was a non-specific error!", e)
finally:
    if file.closed == False:
        file.close()

```

Presentation -----#

```

class IO:

```

```

    """
    This class includes functions related to:
    1- getting user input
    2- displaying user input

```

ChangeLog: (Who, When, What)
J Bailey, 8/19/25, Created Class
J Bailey, 8/19/25, Added menu output and input functions
J Bailey, 8/19/25, Added function to display data
J Bailey, 8/19/25, Added function to display custom error messages
 """

```

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

```

ChangeLog: (Who, When, What)
RRoot, 1.3.2030, Created function
J Bailey, 8/19/25, Imported function from Lab03

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

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

ChangeLog: (Who, When, What)
RRoot, 1.3.2030, Created function
J Bailey, 8/19/25, Imported function from Lab03

```
:return: None
"""
```

```
print() # Adding extra space to make it look nicer.
print(menu)
```

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

ChangeLog: (Who, When, What)
J Bailey, 8/19/25, Create function

```
:return: User menu choice
"""
```

```
choice = "0"
try:
    choice = input("Enter your menu choice number: ") # note this for the next lab
    if choice not in ('1', '2', '3', '4'):
        raise Exception("Please, choose only 1, 2, 3, or 4")
except Exception as e:
    IO.output_error_message(e.__str__()) # Not passing e to avoid the technical message

return choice
```

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

ChangeLog: (Who, When, What)
J Bailey, 8/19/25, Created function

```
:return: dictionary of student data
"""
```

```

try:
    # Input the data
    student_first_name = input("What is the student's first name? ")
    if not student_first_name.isalpha():
        raise ValueError("The first name should not contain numbers.")

    student_last_name = input("What is the student's last name? ")
    if not student_last_name.isalpha():
        raise ValueError("The last name should not contain numbers.")

    course_name = input("What is the course name? ")
    if course_name=="":
        raise ValueError("The course name should not be blank.")

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

@staticmethod
def output_student_data(student_data: list):
    """ This function prints the first name, last name, and course name
    for students entered by the user
    ChangeLog: (Who, When, What)
    J Bailey, 8/19/25, Created function
    """
    print("-" * 50)
    for student in students:
        print(f'{student["FirstName"]},{student["LastName"]}, '
              f'{student["CourseName"]}')
    print("-" * 50)

# End of function definitions

# Main body of script -----#

# When the program starts, read the file data into a list of lists (table)
students = FileProcessor.read_data_from_file(
    file_name=FILE_NAME, student_data=students)

# Begin user interaction
while True:
    #Present menu of options and receive user choice
    IO.output_menu(menu=MENU)
    menu_choice = IO.input_menu_choice()

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

```

```
#Present current student data
elif menu_choice == "2":
    IO.output_student_data(student_data=students) # Added this to improve user experience
    continue

#Save data to JSON file and display data saved to user
elif menu_choice == "3":
    FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
    continue

#Exit program
elif menu_choice == "4":
    break # out of the while loop

else:
    print("Please only choose option 1, 2, 3, or 4")

print("Program Ended")
```

Appendix 2. PyCharm output.

```
/Users/jen/PycharmProjects/PythonProject/.venv/bin/python
/Users/jen/Desktop/Foundations_of_Python/_Module06/Assignment/Assignment06.py
```

---- 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

What is the student's first name? Lisa

What is the student's last name? Simpson

What is the course name? Band

You have registered Lisa Simpson for Band.

---- 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

What is the student's first name? Bart

What is the student's last name? Simpson

What is the course name? Remedial Math

You have registered Bart Simpson for Remedial Math.

---- 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

Bob,Smith,Python 100

Sue,Jones,Python 100

Lisa,Simpson,Band

Bart,Simpson,Remedial Math

---- 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

Data written to Enrollments.json:

Bob Smith Python 100

Sue Jones Python 100

Lisa Simpson Band

Bart Simpson Remedial Math

---- 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

Appendix 3. Output from MacOS Terminal.

```
[jen@Reggies-MacBook-Pro Assignment % python3 Assignment06.py
```

```
----- 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
What is the student's first name? Homer
What is the student's last name? Simpson
What is the course name? Nuclear Physics
You have registered Homer Simpson for Nuclear Physics.
```

```
----- 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
What is the student's first name? Marge
What is the student's last name? Simpson
What is the course name? Home Econ
You have registered Marge Simpson for Home Econ.
```

```
----- 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
-----
```

```
Bob,Smith,Python 100
Sue,Jones,Python 100
Lisa,Simpson,Band
Bart,Simpson,Remedial Math
Homer,Simpson,Nuclear Physics
Marge,Simpson,Home Econ
-----
```

```
----- 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
Data written to Enrollments.json:
Bob Smith Python 100
Sue Jones Python 100
Lisa Simpson Band
Bart Simpson Remedial Math
Homer Simpson Nuclear Physics
Marge Simpson Home Econ
```

```
---- 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
```

```
jen@Reggies-MacBook-Pro Assignment %
```