

Assignment 5 for Foundations of Python Programming

Charles J Lloyd, Student, University of Washington

11/10/2025

Introduction

This document describes the development and testing of a Python script demonstrating the use of dictionaries, .json files, and exception handling in addition to the capabilities demonstrated in Assignment 4.

This assignment was completed using the PyCharm IDE (version 2025.2.3) and Python Version 3.14.0 (64-bit). The GitHub platform was used to distribute this script and related documents.

This work was completed in fulfillment of the requirements for one of ten homework assignments for the course IT FDN 110 A Au 25: Foundations of Python Programming. This course was offered by the University of Washington (UW) department of Professional & Continuing Education. The instructors for this asynchronous on-line course include:

- Randal Root, Instructor
- Becky Peltz, Instructional Assistant

All files associated with this course were developed and tested on a Dell Inspiron 16 Plus 7620 laptop computer running Microsoft Windows 11 Home, Version 10.0.26100 Build 26100. The files were stored locally in the directory C:/Python/PythonCourse.

Procedure

Prior to working on the coding portion of the assignment, the student read the notes and viewed the videos indicated in the documents “Mod05-Notes.docx” and “Mod05-Assignment.docx”, provided for this course on the UW Canvas site. Section 4 of this

document contains the “Acceptance Criteria” for this assignment which essentially make up the “Project Specification.”

The steps involved in creating this program included:

- Reviewed the project specification
- Copied and updated the standard header created for previous assignments
- Added constants and variables required by the project specification
- Completed/updated code for each menu choice
- Tested code for many combinations of menu choice, with and without the Enrollments file in the default directory
- Created account on the GitHub platform
- Created new repository named Python100course and uploaded the Python script
- Updated the external documentation (this document)
- Uploaded all files to the GitHub repository

Program Organization

This program first creates the menu as a string constant with four options:

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

After setting the filename as a constant and declaring the primary variables, the program attempts to open the data file. A try-except structure was used to manage errors that might occur during this process:

```
try:  
    file = open(FILE_NAME, "r") # Open the JSON file for reading  
    students = json.load(file)  
    file.close()  
except FileNotFoundError as e:  
    print("-- Technical Error Message -- ")  
    print(e, e.__doc__)  
    print('Data file does not exist, file will be created after saving')  
except Exception as e:
```

```

print("There was a non-specific error!\n")
print("-- Technical Error Message -- ")
print(e, e.__doc__, type(e), sep='\n')
finally:
    print("closing file")
    if file.closed == False:
        file.close()

```

The overall flow of control within the program is managed using a “match-case” structure as indicated in pseudo-code:

```

Print('\n' + MENU)
menu_choice = input('Enter choice, 1 to 4: ')
match menu_choice:
    case '1':    # Register a Student for a Course
    case '2':    # Show current data
    case '3':    # Save data to file
    case '4':    # Exit the program
    case -:      # Handle invalid menu selection inputs

```

Menu Option 1: Register a Student for a Course

When this option is selected the program queries the user to enter the first and last names and the name of the course. After entry, these inputs are presented to the user who must indicate the data were entered correctly before they are accepted by the program. A “[while loop](#)” is used to manage this interaction which repeats until the user responds with a “Y” when asked to confirm accuracy.

Within this loop a try-except structure is used to manage errors that may occur including the error of entering numeric instead of alphabetic characters for the names. After data accuracy is validated by the user, the name data are used to create a dictionary for each student, and this dictionary is appended to the students list of dictionaries as follows:

```

student_data = {"name_first": student_first_name,
                "name_last": student_last_name,
                "name_course": course_name}

students.append(student_data)

```

Menu Option 2: Show current data

When this option is selected the program prints the data for review. If no data exist the “No Data has been entered yet” message is displayed. Any existing data, read from the file and recently entered, is printed using the commands:

```
for student_data in students:
    print(f'{student_data["name_first"]},{student_data["name_last"]},{'
          f'{student_data["name_course"]}]')
```

Menu Option 3: Save data to a file

When this option is selected the program saves the student data to the “Enrollments.csv” file. The list data are saved as comma separated strings. An “if-else” structure is used to avoid creating an empty file in the event no data are available. Within this loop a try-except structure is used to manage any errors that may occur.

Menu Option 4: Exit the program

Upon selection of this option the program **breaks** out of the match-case structure that manages the menu selections after presenting the message “**Exited the program**”

Invalid Menu Selections

The program handles the case of an invalid menu selection inputs (any keys other than 1 to 4) using a **continue** statement that returns program control to the menu (match statement) after presenting the message “**Invalid Menu Choice.**”

Test Results

The Python script in the file “Assignment05.py” was tested in three ways using: the PyCharm IDE, a Windows terminal, and Notepad. Prior to testing the Enrollments file was deleted using Windows.

Initial testing was conducted by running the program from PyCharm. Running the script with the missing data file produced the following messages:

```
-- Technical Error Message --  
[Errno 2] No such file or directory: 'Enrollments.json' File not found.  
Data file does not exist, file will be created after saving
```

followed by presentation of the menu. The second line of the message was produced by Python while the last line was provided to provide the user with more specific guidance.

Menu option 2 was then selected which produced the “No data have been entered yet” message as expected.

Menu option 1 was selected and the first, last, and course names were entered and validated as they were in the previous assignment:

```
First Name Entered: Charles  
Last Name Entered: Lloyd  
Course Name Entered: Python100  
Enter "Y" if correct, or any other key to re-enter: y
```

Menu option 2 was then selected which produced:

```
Student registration data:  
Charles,Lloyd,Python100
```

Menu option 3 was selected which saved the data creating the Enrollments.csv file, presented the message “Saved the records to file”, and printed the data that would go to the file:

```
{'name_first': 'Charles', 'name_last': 'Lloyd', 'name_course': 'Python100'}
```

After exiting the program using option 4, the contents of the Enrollments file were examined using Notepad and found to contain a single record, in the .json file format, as expected.

The program was restarted, and the data for three additional records was entered using option 1 and saved using option 3. The set of records was examined using option 2:

```
Student registration data:  
Charles,Lloyd,Python100  
Amelia,DeLoach,Java  
Charlie,Lloyd,Mechanics  
Zoe,Lloyd,Algebra
```

The program was stopped using option 4 and the content of the Enrollments file was examined using Notepad and found to contain the four records be as expected:

```
[
  {
    "name_first": "Charles",
    "name_last": "Lloyd",
    "name_course": "Python100"
  },
  {
    "name_first": "Amelia",
    "name_last": "DeLoach",
    "name_course": "Java"
  },
  {
    "name_first": "Charlie",
    "name_last": "Lloyd",
    "name_course": "Mechanics"
  },
  {
    "name_first": "Zoe",
    "name_last": "Lloyd",
    "name_course": "Algebra"
  }
]
```

Many of the tests described above were repeated after deleting the Enrollments file and running the program from a Windows terminal by setting the path:

```
cd c:\Python\PythonCourse\_Module05\Assignment
```

And entering the command:

```
python Assignment05.py
```

The results were identical to those obtained using the PyCharm IDE.

Distribution

The following three files were uploaded to the GitHub repository:

- Assignment05.py
- Assignment05_Description_CharlesLloyd.pdf
- Enrollments.json

The link for this repository is provided at the top of this document.