

Elise C.

2/25/2026

Foundations of Python Programming

Assignment 05

GitHubURL

## Advanced Collections and Error Handling

### Introduction

This week I learned to build a more complex Python program utilizing last week's assignment as a base file. This document will detail my process in creating a Python program that included the while loop, programming menus, and conditional logic. In addition, Assignment 05 adds the use of data processing using dictionaries, json files and exception handling.

### Preparing to Write

First, I created a new project called "Module 05" in PyCharm IDE. Then I opened the starter file with the instructor's code from Assignment 04. I saved the file as instructed by the homework document and revised the script header. I utilized the notes from the instructor to help format my scripts.

### Starting the Writing of Program

This week I created a new dictionary variable. At first it took some time to compare the code for csv files and for json files. It was important to learn how the script should be modified to interact successfully with a json file. I needed to modify the code to bring the json data into the program and then save the new student entries back to the json file.

### Testing and Modifying the Code

Once I had reviewed the code a few times and it looked ready, I ran the program. All the menu options ran smoothly. I checked the json file and the one new student had been added to the file. The formatting did not look great; it was one long row. At this point I had three students in the json file. I decided to refer back to the module 05 notes and realized that I could change to the code to format the json file better. On line 112 I added 'indent=2' to the code here: "json.dump(students, file, indent=2)." When I reviewed other example code files from the

course, I also noticed that my indentation was a bit different so I changed that as well on lines 80 and 81. The new code adds one indentation to the “LastName” and “CourseName” lines:

```
student_data = {"FirstName": student_first_name,  
                "LastName": student_last_name,  
                "CourseName" : course_name}
```

Once my code was updated, I ran the program again in PyCharm and I entered one more student (see Figures 1 and 3). I checked the json file and all the data had been reformatted into a much more readable file. When I ran it in the terminal, it worked well the first time. In the terminal I entered two students to confirm that my program worked for multiple entries (see Figure 3). I also tested the error handling using PyCharm. I moved the Enrollments.json file into another folder and got the “JSON file must exist before running this script!” Then I tested entering numbers in the first name and then the last name prompt. I received the “name should not contain numbers” error. The error testing went smoothly.

## Final Program

After making changes in my code to fix the formatting in the json file, I created a program that ran well in PyCharm (Figure 1) and the terminal (Figure 2). I was also able to add two new students to my json file (see Figure 3).

Figure 1: PyCharm Output Entering the Second New Student

---- 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.
- 

What would you like to do: 1

Enter the student's first name: Larry

Enter the student's last name: Larson

Please enter the name of the course: 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.
- 

What would you like to do: 2

You have registered Larry Larson for Python 100.

---

Bob,Smith, Python 100

Sue,Jones, Python 100

Elise,Christiansen, Python 100

Larry,Larson, 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.
-

What would you like to do: 3

The following data was saved to file!

Bob,Smith, Python 100

Sue,Jones, Python 100

Elise,Christiansen, Python 100

Larry,Larson, 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.
- 

What would you like to do: 4

Program Ended

Process finished with exit code 0

## Reflections

This was my first time working with a json file and creating ways to handle errors within the code. This assignment was helpful in developing more useful skills regarding managing data and errors. I will be also creating my first account on GitHub.

Figure 2. Command Prompt Output Entering Two Students' Data

```
Command Prompt      X + ▾
Microsoft Windows [Version 10.0.26200.7840]
(c) Microsoft Corporation. All rights reserved.

C:\Users\elise>CD Documents\Python\PythonCourse

C:\Users\elise\Documents\Python\PythonCourse>Python Assignment05.py
FirstName: Bob, LastName: Smith, CourseName: Python 100
FirstName: Sue, LastName: Jones, CourseName: 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.
-----
What would you like to do: 1
Enter the student's first name: Elise
Enter the student's last name: Christiansen
Please enter the name of the course: 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.
-----
What would you like to do: 2
You have registered Elise Christiansen for Python 100.
-----
Bob,Smith, Python 100
Sue,Jones, Python 100
Elise,Christiansen, 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.
-----
What would you like to do: 3
The following data was saved to file!
Bob,Smith, Python 100
Sue,Jones, Python 100
Elise,Christiansen, 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.
-----
What would you like to do: 1
Enter the student's first name: Larry
Enter the student's last name: Larson
Please enter the name of the course: 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.
-----
```

```
What would you like to do: 2
You have registered Larry Larson for Python 100.
-----
```

```
Bob,Smith, Python 100
Sue,Jones, Python 100
Elise,Christiansen, Python 100
Larry,Larson, Python 100
-----
```

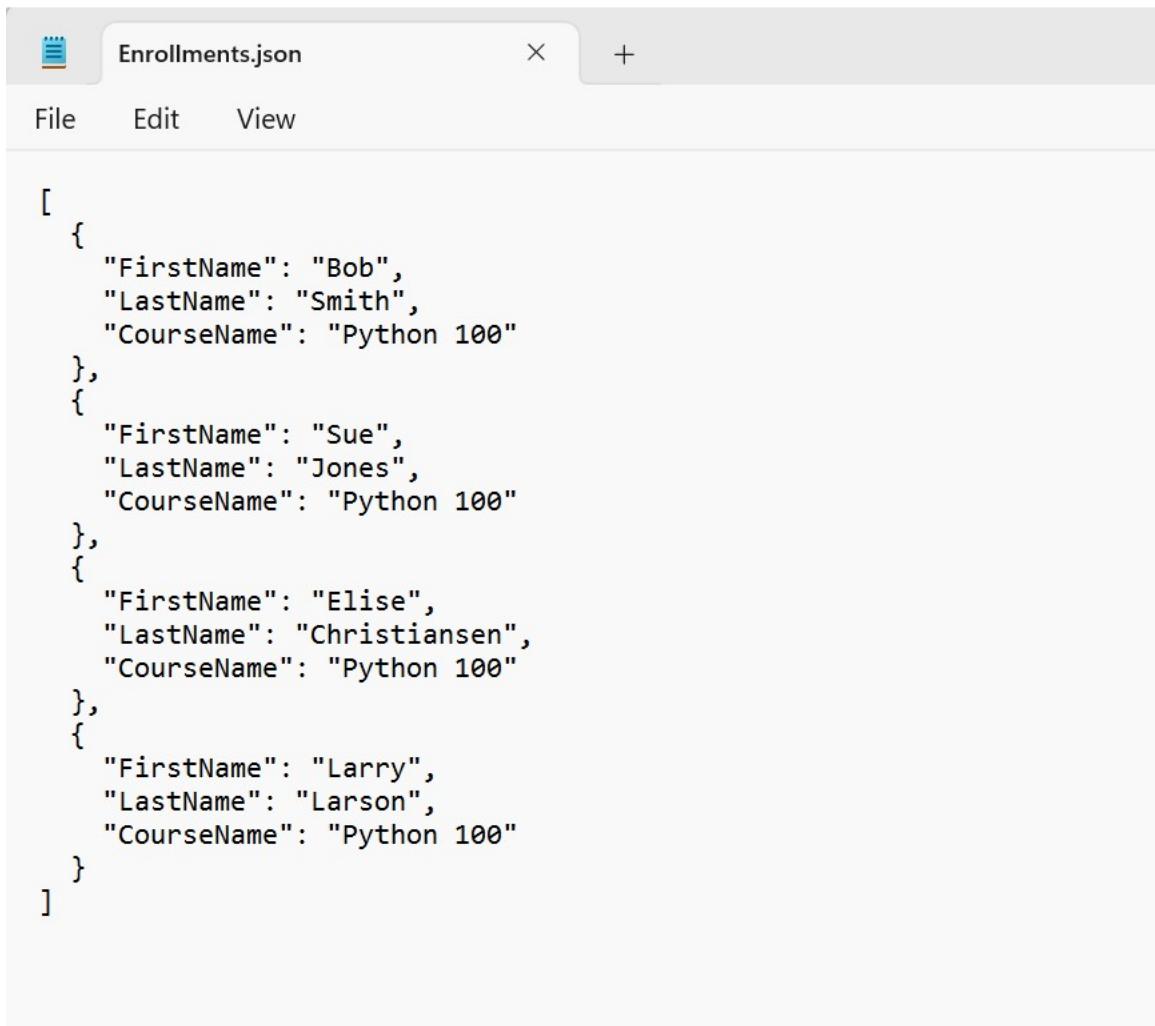
```
---- Course Registration Program ----
Select from the following menu:
 1. Register a Student for a Course.
 2. Show current data.
 3. Save data to a file.
 4. Exit the program.
-----
```

```
What would you like to do: 3
The following data was saved to file!
Bob,Smith, Python 100
Sue,Jones, Python 100
Elise,Christiansen, Python 100
Larry,Larson, 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.
```

```
-----  
What would you like to do: 4  
Program Ended  
C:\Users\elise\Documents\Python\PythonCourse>
```

Figure 3. Notepad JSON File with Four Students



The screenshot shows a Notepad window titled "Enrollments.json". The window has a standard menu bar with "File", "Edit", and "View" options. The main content area displays a JSON array with four elements, each representing a student enrollment:

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

## Summary

This assignment allowed me to create a more complex Python program with a json file, error handling, and a dictionary. It was interesting to compare the code differences when working with csv files and json files. I look forward to learning more about error handling.