

Jiahua (Joey) Zhang

5/21/2025

IT FDN 110 A

Assignment 05

[GitHub URL](#)

Advanced Collections and Error Handling with Knowledge Check

Introduction

This module focuses on advanced collections using dictionaries, and Javascript Object Notation (JSON) files as well as applying error handling concepts to the code. It begins with writing data to and from a file using dictionaries and delves into the benefits of JSON files and stresses the importance of structured error handling and the try-except construct. Finally, module 5 discusses the importance of GitHub in collaborative software development. There is a Python script exercise below that highlights all the concepts that I learned reading module 5.

Dictionary Collections

Like a table of data using a list of lists, a dictionary table is created by using a “table”. However, there are some key differences between a list and a dictionary where dictionaries have keys and values where the column names are the keys, and the rows of data are the values in those columns. This is basically how a row of data in a spreadsheet is formatted. Other important concepts are the keys in a dictionary are case sensitive and double quotes are used around the keys’ names. JSON files as mentioned above also use this format and are related to dictionaries.

An example of a dictionary is provided (module 05, page 3):

```
row1 = {"ID":1,"Name":"Bob Smith", "Email":"BSmith@Notmail.com"}
row2 = {"ID":2,"Name":"Sue Jones", "Email":"SueJ@Yayoo.com"}
table = [row1, row2]
```

“ID” is the key and integers 1 and 2 in each row are values.

To add a new row of data, you can append the additional data to a table using the “table.append(row)” code. Similarly, to remove data from a table, one needs to include code that searches the table for a matching ID (e.g. if int(id) == row[“ID”]:) and use the “table.remove(row)” code to update the table.

JSON Files

JSON (Javascript Object Notation) files are lightweight data-interchange formatted files that is easy for both users to read and write and for computers to parse and generate. It is mainly used for client/server communication, file configuration and data storage. JSON files are very similar to Python dictionaries where they consist of key-value pairs. The keys are strings enclosed in double quotations and data is organized by using curly braces {} for objects and square brackets [] for arrays which are ordered lists of values within a key. JSON allows nested objects and arrays within objects and arrays which allows the construction of complex data structures.

The key differences between a JSON file and a CSV file are CSV files are simpler tabular formatted files used for large data storage, import/export functions, and quick analysis and JSON files tend to be larger than CSV files and are hierarchical formatted files used for configuration, web APIs, and document-oriented databases.

There is a built-in JSON module within Python that painlessly allows one to write Python dictionaries to a JSON file using the “json.dump(table, file)” function.

Structured Error Handling (Try-Except)

Error handling improves error management of one’s scripts by actively planning for mistakes ahead of time and making the debugging process a lot easier. This is done through the try-except construct in Python which generates user-friendly (customizable) error messages than the technical explanations the program outputs when the code runs into errors. See example of a Try-Except statement below (Module 05, Page 20):

```
try:
    value1: float = 3.0
    value2: float = 0.0
    print(value1 / value2)
except Exception as e:
    print("Error! Please check you are not dividing by zero.\n")
    print("-- Technical Error Message -- ")
    print(e)          # Print the exception object (typically includes the error
message)
    print(type(e))    # Print the type of the exception object
    print(e.__doc__)  # Print the documentation string of the exception type
    print(e.__str__()) # Print the string representation of the exception
```

This example of an error message uses a combination of simple and complex messages that allow users with different skill levels to access.

There are also specific exceptions you can use to catch bugs such as “ZeroDivisionError as e” and “FileNotFoundError as e”. The “Exception as e” is used as a catch all for any other exceptions that may occur.

Another way to handle errors is by raising custom errors. Some common practices of this are using the “if new_file_name[0].isnumeric():” and “if not student_last_name.isalpha()” functions which signals the program to warn the user if the file name begins with a number and if the student’s last name is not alphabetical.

Finally, the exception block not only can display error messages, but it can also be used to avoid errors later in the code such as making sure the file closes after it is opened.

Managing Code Files and GitHub

Saving and sharing code are best practices that help foster collaboration, code quality and integrity, knowledge sharing and software development efficiency. The most important areas that I have seen this used in my work are backup and disaster recovery, documentation and knowledge transfer for new hires, and code reusability.

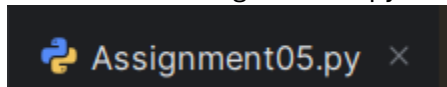
There are two ways that developers can share their code: network and cloud sharing. My work used to use network sharing through network paths that all team members have access to and have currently moved on to cloud sharing such as using Microsoft Teams/One Drive.

GitHub is a cloud-based platform for file hosting, code repository hosting, version control, access control and collaboration features. I have created a GitHub account in the past and will be using it again for this course.

Assignment05.py Knowledge Check

File Name

File is named Assignment04.py to signify that this is a python program.



Script Header

For proper documentation, I have included the following header to display the title, description, and change log to my name as the creator of the script along with the date.

```
# ----- #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   JZhang, 5/13/2025, Created Script
#   JZhang, 5/21/2025, Modified Script
# ----- #
```

Body

The first step in assignment05 is to build off assignment04 and use JSON dictionaries as well as try-except functions to add more checks to the code.

Following the instructions, I updated the FILE_NAME to a JSON file and changed student_data to a dictionary when it was previously set to a list. I also removed csv_data: str = '' since CSV is not needed with a JSON file.

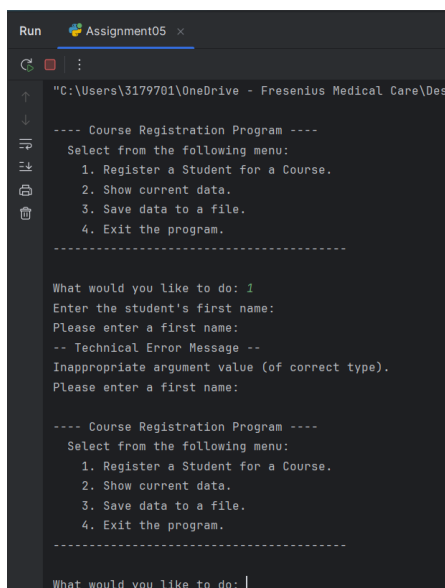
The instructions also specify that the existing Enrollments.json file be read upon running the program, so I added code to open and read the file and to extract the data from the existing file and transform the data from the file to be loaded into my collection using the "json.dump()" function.

Additionally, I added Try-Except functions as stated in the assignment error-handling instructions to provide structured error handling messages when the file is read into the list of dictionary rows, when the user enters a first name, when the user enters a last name and when the dictionary rows are written to the file.

The last step is to print the file data so the user can see the displayed data and the loop ends stopping the program.

Test

Confirm that the program takes and displays the user's inputs and saves it on top of the existing data in Enrollments.json where multiple registrations are allowed. I have also tested the Try-Except error handling to confirm that my custom error messages pop up when I purposely left the first name blank and included a number in the first name the second run (see below):



```
Run Assignment05 x
"C:\Users\3179701\OneDrive - Fresenius Medical Care\Desktop\Assignment05.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.

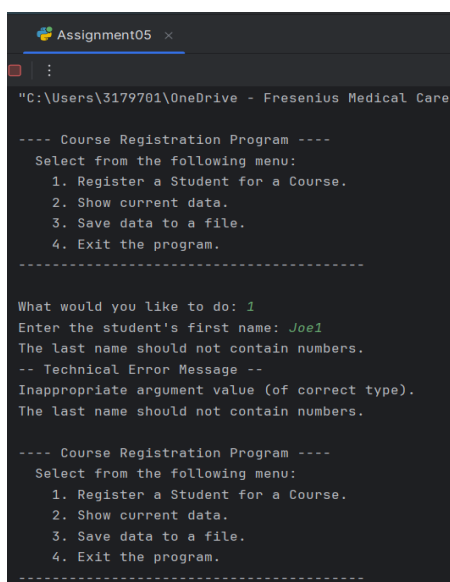
-----

What would you like to do: 1
Enter the student's first name:
Please enter a first name: --
-- Technical Error Message --
Inappropriate argument value (of correct type).
Please enter a first name:

---- 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: |
```



```
Assignment05 x
"C:\Users\3179701\OneDrive - Fresenius Medical Care\Desktop\Assignment05.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.

-----

What would you like to do: 1
Enter the student's first name: Joe1
The last name should not contain numbers.
-- Technical Error Message --
Inappropriate argument value (of correct type).
The last name should not contain numbers.

---- 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.

-----
```

PyCharm Test with JSON output

Before running code:

```
{ } Enrollments.json X
C: > Users > 3179701 > OneDrive - Fresenius Medical Care > Desktop > Python > PythonCourse > { } Enrollments.json > ...
1 [
2  {"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"},
3  {"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"}
4 ]
```

After running code:

```
{ } Enrollments.json C:\PythonCourse X { } Enrollments.json C:\A05 •
C: > Users > 3179701 > OneDrive - Fresenius Medical Care > Desktop > Python > PythonCourse > A05 > { } Enrollments.json
1 [
2  {
3    "FirstName": "Bob",
4    "LastName": "Smith",
5    "CourseName": "Python 100"
6  },
7  {
8    "FirstName": "Sue",
9    "LastName": "Jones",
10   "CourseName": "Python 100"
11  },
12  {
13   "FirstName": "Joey",
14   "LastName": "Zhang",
15   "CourseName": "Python 100"
16  },
17  {
18   "FirstName": "Ken",
19   "LastName": "Chang",
20   "CourseName": "Python 100"
21  }
22 ]
```

Command Terminal Test (see next page):

cmd Command Prompt

Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\3179701>CD C:\Users\3179701\OneDrive - Fresenius Medical Care\Desktop\Python\PythonCourse\A05

C:\Users\3179701\OneDrive - Fresenius Medical Care\Desktop\Python\PythonCourse\A05>python Assignment05.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.  
-----
```

```
What would you like to do: 1  
Enter the student's first name: Joey  
Enter the student's last name: Zhang  
Please enter the name of the course: Python 100  
You have registered Joey Zhang 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.  
-----
```

```
What would you like to do: 2  
-----
```

```
Student Bob Smith is enrolled in Python 100  
Student Sue Jones is enrolled in Python 100  
Student Joey Zhang 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.  
-----
```

```
What would you like to do: 1  
Enter the student's first name: Ken  
Enter the student's last name: Chang  
Please enter the name of the course: Python 100  
You have registered Ken Chang 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.
-----

What would you like to do: 2
-----
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Joey Zhang is enrolled in Python 100
Student Ken Chang 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.
-----

What would you like to do: 3
The following data was saved to file!
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Joey Zhang is enrolled in Python 100
Student Ken Chang 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.
-----

What would you like to do: 4
Program Ended

C:\Users\3179701\OneDrive - Fresenius Medical Care\Desktop\Python\PythonCourse\A05>
```

Summary

The fifth module of this Introduction to Python course teaches the student how to properly work with dictionaries and Try-Except functions to present data in tables into a JSON file and catch errors in advance. It also illustrates the importance of sharing code and collaborating with each other to ensure code quality and improve knowledge sharing.