

Tulip O'Neill

May 21, 2025

Foundations of Programming

Assignment 05

05: Advanced Collections & Error Handling

Introduction

Over the course of the last week, I have learned how to use dictionaries with file data and how to incorporate JSON files and exceptions for error handling into my code. I found that these concepts were a little more straightforward to me, maybe because they are building off of skills I have already started to learn. The error handling was a little confusing at first, but implementing exceptions into my code during demo03 and lab03 helped greatly. All of this has come together to provide me the tools to create my script for this week's assignment.

In this paper, I will be sharing how I created a program that presents a menu and provides information about a student's enrollment. This is similar to previous weeks but the new script adds the use of data processing using dictionaries and exception handling.

Drafting the Script

I began drafting my script by copying over the starter file provided and making minor adjustments to the backbone of the code. The more edits I started to make, the more confused I got with what was supposed to be included in the code and what wasn't. Having these starter files is really confusing to me and I would prefer to either start fresh or work off of my own previous work. I decided to keep working off of the starter file because understanding what other people's intentions are with their code is will likely be very valuable. What was initially confusing to me was what from the CSV portions needed to be carried over to the JSON portions. To make this all a little more manageable, I broke it into two parts: updating the script to account for the use of JSON rather than CSV, and error handling. Once I drafted my script with everything needed to run with the JSON file, I tested it and was only met with minor errors. The errors I encountered were "KeyErrors" because I had named the keys something different than what was provided in the enrollments.json file. Once the keys in the JSON file were updated, the code ran smoothly.

```

11 # When the program starts, read the file data into a list of lists (table)
12 file = open(FILE_NAME, "r")
13 students = json.load(file)
14 file.close()
15
16 # Present and Process the data
17 while True:
18
19     # Present the menu of choices
20     print(MENU)
21     menu_choice = input("Choose a menu option (1,2,3,4): ")
22
23     # Input user data
24     if menu_choice == "1": # This will not work if it is an integer!
25         print("To register a student, please enter the following information...")
26         student_first_name = input("Student first name: ")
27         student_last_name = input("Student last name: ")
28         course_name = input("Course name: ")
29         student_data = {"first Name": student_first_name,
30                        "last Name": student_last_name,
31                        "Course": course_name}
32         students.append(student_data)
33         print(f"{student_first_name} {student_last_name} has successfully been registered for {course_name}.")
34         continue
35
36     # Present the current data
37     elif menu_choice == "2":
38
39         # Process the data to create and display a custom message
40         print("-" * 50)
41         for student in students:
42             print(f"{student['first Name']}, {student['last Name']}, {student['Course']}")
43
44     elif menu_choice == "3":
45         # Save the data to a file
46         elif menu_choice == "3":
47             file = open(FILE_NAME, "w")
48             json.dump(students, file)
49             file.close()
50             print("-" * 50)
51             for student in students:
52                 print(f"{student['first Name']}, {student['last Name']}, {student['Course']}")
53                 print("-" * 50)
54             continue
55
56     # Stop the loop
57     elif menu_choice == "4":
58         break # out of the loop
59     else:
60         print("Please only choose option 1, 2, 3, or 4")
61
62 print("Program Ended")

```

Figure 1: Draft of script updated to use JSON rather than CSV

```

Choose a menu option (1,2,3,4): 1
To register a student, please enter the following information...
Student first name: Test
Student last name: Tester
Course name: 101
Test Tester has successfully been registered for 101.

---- 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.

-----
Choose a menu option (1,2,3,4): 2
-----
Traceback (most recent call last):
  File "/Users/tulip/Documents/Python/PythonCourse/Module05/Assignment/Assignment05.py", line 65, in <module>
    print(f"{student['first Name']}, {student['last Name']}, {student['Course']}")
    ~~~~~^~~~~~
KeyError: 'first Name'

Process finished with exit code 1

```

Figure 2: Key error due to keys in PyCharm not matching the JSON file

Now that my script was running with the JSON edits, I needed to incorporate the error handling. I started with the first section, which reads the file data. I decided to add a file not found exception and a general exception. On top of this, I learned about an exception that was specific to when an error occurs trying to turn contents of a file to a JSON dictionary, so I included that for fun. The next set of exceptions pertained to the user input for student names. These exceptions were essentially the same as taught in the previous lab, so it was fairly easy to add in. One addition I made was including “continue” at the end of my “if not” and “except” sections. This is because I did not want any bad data to be entered into my list of dictionaries and I wanted users to be able to try again if they entered something that was not allowed. I have an apostrophe in my name, so this is all too familiar haha. The last exception to include was at the end, when writing to the JSON file. This again was very similar to the previous lab. I chose to have the “continue” statement under “finally” because I want the loop to continue whether or not saving the file is completed. After adding this in, my code was complete! I ran the script in PyCharm and the terminal with no errors. Beyond that, only minor formatting changes were made since no major errors occurred.

```

34 # When the program starts, read the file data into a list of dictionaries (table)
35 try:
36     file = open(FILE_NAME, "r")
37     students = json.load(file)
38     file.close()
39 except FileNotFoundError as e:
40     print("Please check that file exists.\n")
41     print("-- Technical Error Message -- ")
42     print(e, e.__doc__, type(e), sep='\n')
43 except json.decoder.JSONDecodeError as e:
44     print("There was an error parsing the file.\n")
45     print("-- Technical Error Message -- ")
46     print(e, e.__doc__, type(e), sep='\n')
47 except Exception as e:
48     print("There was a non specific error.\n")
49     print("-- Technical Error Message -- ")
50     print(e, e.__doc__, type(e), sep='\n')
51 finally:
52     if file.closed == False:
53         file.close()

```

Figure 3: Try/Except for reading the file data

```

62 # Input user data
63 if menu_choice == "1": # This will not work if it is an integer!
64     print("To register a student, please enter the following information...")
65     try:
66         student_first_name = input("Student first name: ")
67         if not student_first_name.isalpha():
68             print("First name must only contain letters. Please try again.")
69             continue
70
71         student_last_name = input("Student last name: ")
72         if not student_last_name.isalpha():
73             print("Last name must only contain letters. Please try again.")
74             continue
75
76     except ValueError as e:
77         print(e) # Prints the custom message
78         print("-- Technical Error Message -- ")
79         print(e.__doc__)
80         print(e.__str__())
81         continue
82     except Exception as e:
83         print("There was a non-specific error!\n")
84         print("-- Technical Error Message -- ")
85         print(e, e.__doc__, type(e), sep='\n')
86         continue

```

Figure 4: Try/Except for user name input

```

106 # Save the data to a file
107 elif menu_choice == "3":
108     try:
109         file = open(FILE_NAME, "w")
110         json.dump(students, file)
111         file.close()
112         print("-" * 50)
113         for student in students:
114             print(f"{student['First Name']},{student['Last Name']},{student['Course']}")
115         print("-" * 50)
116     except TypeError as e:
117         print("Please check that the data is a valid JSON format\n")
118         print("-- Technical Error Message -- ")
119         print(e, e.__doc__, type(e), sep='\n')
120     except Exception as e:
121         print("-- Technical Error Message -- ")
122         print(e, e.__doc__, type(e), sep='\n')
123     finally:
124         if file.closed == False:
125             file.close()
126         continue

```

Figure 5: Try/Except for writing to file

Summary

In conclusion, I was able to use this week's skills to build off of my previous assignment and create a program that presents a menu and provides information about a student's enrollment in a Python course. The key differences included using dictionaries and JSON files as well as including error handling in the form of try/except. Although these changes were fairly minor, our programs are now getting more advanced and smaller changes take more effort in terms of organization and mental understanding. I'm nervous and excited to see what next week brings.

Appendix

Link to GitHub: <https://github.com/tuliponeill/IntroToProg-Python-Mod05>

Final script:

```
# -----  
----- #  
# Title: Assignment05  
# Desc: This assignment demonstrates using dictionaries, files, and exception  
handling  
# Change Log: (Who, When, What)  
#   RRoot, 1/1/2030, Created script  
#   TONeill, 05/21/2025, Created script  
# -----  
----- #  
  
# Import libraries  
import json  
  
# 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  
#csv_data: str = "" # Holds combined CSV data.  
file = None # Holds a reference to an opened file.  
menu_choice: str # Holds the menu choice made by the user.  
  
# When the program starts, read the file data into a list of dictionaries  
(table)  
try:
```

```

    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError as e:
    print("Please check that file exists.\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
except json.decoder.JSONDecodeError as e:
    print("There was an error parsing the file.\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
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:
    if file.closed == False:
        file.close()

# Present and Process the data
while True:

    # Present the menu of choices
    print(MENU)
    menu_choice = input("Choose a menu option (1,2,3,4): ")

    # Input user data
    if menu_choice == "1": # This will not work if it is an integer!
        print("To register a student, please enter the following
information...")
        try:
            student_first_name = input("Student first name: ")
            if not student_first_name.isalpha():
                print("First name must only contain letters. Please try
again.")
                continue

            student_last_name = input("Student last name: ")
            if not student_last_name.isalpha():
                print("Last name must only contain letters. Please try
again.")
                continue

        except ValueError as e:
            print(e) # Prints the custom message
            print("-- Technical Error Message -- ")
            print(e.__doc__)
            print(e.__str__())
            continue
        except Exception as e:
            print("There was a non-specific error!\n")
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')
            continue

    course_name = input("Course name: ")
    student_data = {"First Name": student_first_name,

```

```

        "Last Name": student_last_name,
        "Course": course_name}
    students.append(student_data)
    print(f"{student_first_name} {student_last_name} has successfully
been registered for {course_name}.")
    continue

# Present the current data
elif menu_choice == "2":

    # Process the data to create and display a custom message
    print("-"*50)
    for student in students:
        print(f"{student["First Name"]},{student["Last
Name"]},{student["Course"]}")
    print("-"*50)
    continue

# Save the data to a file
elif menu_choice == "3":
    try:
        file = open(FILE_NAME, "w")
        json.dump(students, file)
        file.close()
        print("-" * 50)
        print("The following data has been saved to file:")
        for student in students:
            print(f"{student["First Name"]},{student["Last
Name"]},{student["Course"]}")
        print("-" * 50)
    except TypeError as e:
        print("Please check that the data is a valid JSON format\n")
        print("-- Technical Error Message -- ")
        print(e, e.__doc__, type(e), sep='\n')
    except Exception as e:
        print("-- Technical Error Message -- ")
        print(e, e.__doc__, type(e), sep='\n')
    finally:
        if file.closed == False:
            file.close()
        continue

# Stop the loop
elif menu_choice == "4":
    break
else:
    print("Please only choose option 1, 2, 3, or 4")

print("Thank you for using this program.")

```