

Name: Mike Luke

Date: 11/10/2025

Course: IT FDN 110 A Au 25: Foundations Of Python Programming

Assignment 05 –

Introduction

For Week 5, we're introduced to json files, dictionaries, and Error Handling. This caused a bit of trouble due to the different formatting of the json files. It required the use of the Python's json module to open the file. Using the dictionary function you had to map variables to a key, which would allow Python to tie two pieces of data together in the program.

Topic

Beginning to work through the assignment, the first issue I ran into was attempting to create a dictionary. I did this in my assignment 3 but I struggled with understanding how it differs from what I submitted in that assignment.

```
course_menu = """
--- Choose your Course ---
1. Python 100
2. C++ 100
3. Java 100
4. PHP 100
5. React 100
-----
"""

print(course_menu)
course_choice = input('Enter your choice 1-5: ')

# Map course choice to course name
course_options = {
    '1': 'Python 100',
    '2': 'C++ 100',
    '3': 'Java 100',
    '4': 'PHP 100',
    '5': 'React 100'
}

if course_choice in course_options:
    enter_course = course_options[course_choice]
    print(f'{student_first_name}, {student_last_name}, {enter_course}')
    csv_data = (f'{student_first_name},{student_last_name},'
                f'{enter_course}\n')
else:
    print('Invalid choice')
```

Assignment 3 submission for dictionary

```
# Input user data
if menu_choice == "1": # This will not work if it is an integer!
    student_first_name = input("Enter the student's first name: ")
    student_last_name = input("Enter the student's last name: ")
    course_name = input("Please enter the name of the course: ")
    # Add Dictionary New Row
    student_data = {'FirstName': 'student first name',
                    'LastName': 'student last name',
                    'CourseName': 'course name'}
    students.append(student_data)
    print(f"You have registered {student_first_name} {student_last_name} {course_name}.")
    continue
```

Initial Assignment 5 attempt

```
[
  {
    "FirstName": "Bob",
    "LastName": "Smith",
    "CourseName": "Python 100"
  },
  {
    "FirstName": "Sue",
    "LastName": "Jones",
    "CourseName": "Python 100"
  },
  {
    "FirstName": "student_first_name",
    "LastName": "student_last_name",
    "CourseName": "course_name"
  }
]
```

What I realized it 1, it did not input the users data and 2 is changed the formatting of the json file. I realized yet again, I was incorrectly formatting the variable creating a string. Also, the Keys were already defined in the json file (i.e. FirstName, LastName, CourseName). I just need to associate to the input variables that collected the user data. My other menu I set all those values and set constrained input associated with keys.

Next reading through the project and keeping in mind that we were going to have to use error handling, I went ahead and formatted using the try statement with if not and except. This was a bit familiar because on assignment 3 I believe I went down a programming rabbit hole and implemented or attempted to implement error handling. Ultimately, I had to scrap that whole project and start over because I was in the middle of spaghetti code and was struggling to fully walk through the steps in my code I cooked up. I was able to add additional error handling for the sub menu to select the course, which was cool. That took a

bit of work because of indentation issues, so I had to walk through the code to find those mistakes.

```
try:
    student_first_name = input("Enter the student's first name: ")
    if not student_first_name.isalpha():
        raise ValueError('First name must be alphanumeric')
    student_last_name = input("Enter the student's last name: ")
    if not student_last_name.isalpha():
        raise ValueError('Last name must be alphanumeric')
    course_menu = """
    --- Choose your Course ---
    1. Python 100
    2. C++ 100
    3. Java 100
    4. PHP 100
    5. React 100
    -----
    """

    print(course_menu)
    course_choice = input('Enter your choice 1-5: ')

    # Map course choice to course name
    course_options = {
        '1': 'Python 100',
        '2': 'C++ 100',
        '3': 'Java 100',
        '4': 'PHP 100',
        '5': 'React 100'
    }

    if course_choice in course_options:
        course_name = course_options[course_choice]
```

I added an additional message to remind the user to save their data using option 3. I thought this would be a nice touch since as the professor stated it is just held in the list but no actual changes have occurred until its saved. Basically same concept as any program we use with a GUI (Word, Excel, PDF, etc), we type or add information, but if it isn't saved and unexpectedly shuts down or crashes we lose all that data in memory because its not stored on the hard drive.

Choice 2 was straight forward, instead of iterating on rows like in the previous assignment I just needed to iterate over the student data (i.e. row) in students (list of list) and ensure student data variables (i.e. student_first_name, etc) was associated with the key (i.e. Firstname, LastName, CourseName). I also liked the addition of the dash to present the data with separators.

```
# Present the current data
elif menu_choice == "2":
    print('The Following Students are Currently Registered')
    print('-' * 50)
    for student_data in students:
        student_first_name = student_data['FirstName']
        student_last_name = student_data['LastName']
        course_name = student_data['CourseName']
        print(f'{student_first_name}, {student_last_name}, {course_name}')
    print('-' * 50)
    continue
```

Menu 3, json makes it extremely easy to write all the data with a simple json.dump. I added indent=4 for pretty print.

```
{
  "FirstName": "Bob",
  "LastName": "Smith",
  "CourseName": "Python 100"
},
{
  "FirstName": "Sue",
  "LastName": "Jones",
  "CourseName": "Python 100"
},
{
  "FirstName": "Mike",
  "LastName": "Luke",
  "CourseName": "Python 100"
},
{
  "FirstName": "Michael",
  "LastName": "Luke",
  "CourseName": "C++ 100"
},
{
  "FirstName": "Mike",
  "LastName": "Luke",
  "CourseName": "Java 100"
},
}
```

Summary

I enjoyed this assignment, it helped me understand the error handling process. This was something I was trying to figure out a little too early in the course. I thought it would be useful to help limit user errors i.e. isalpha or isdigit or no file, etc, while giving the user adequate information on the error. It makes the whole user experience more enjoyable if you can obtain details of why the program stopped working or there was an issue while using the program. Otherwise, you would most likely continue to run, exit, run, exit and become frustrated with it not knowing what the issue is. The dictionaries allow the developer to reduce chances for human errors. Example being the course name, if you

were to have the user input the course name you would have all kinds of typos or incorrect class titles etc. This helps manage data flow efficiently from input to output for database use elsewhere.