Student Name - Roli Pathak

Class – IT FDN 110 A Wi 25: Foundations Of Programming: Python

Date – 2/26/2025

Assignment: Assignment 05

GitHub link - https://github.com/rolipathak/IntroToProg-Python-Mod05.git

Introduction: This module covers advanced topics in python like dictionaries, file operations like read write and close on JSON files, user input validation and error/exception handling.

Dictionaries:

These are advanced built in data structure in Python. They are used to store key -values which can be string, numbers, Booleans, lists and tuples or other dictionaries. Dictionaries are defined using curly brackets (just like lists are defined in square brackets [])

Example:

```
student_data: dict = {}    # initializing a dictionary

student_data = {"FirstName": student_first_name,   # storing key values in a dictionary
 "LastName": student_last_name,
 "CourseName": course_name}
```

JSON Files: JSON stands for Java Script Object Notation. It's a lightweight data file format just like txt file which can be easy for both people and computers to read and parse through. JSON is a text based file that can store key-value pairs just like dictionaries in a simple and organized way. JSON files can store strings, numbers, Boolean values. JSON files store strings in curly brackets and arrays in square brackets. JSON allows for nesting arrays within objects which can result in more complex data structures.

JSON vs. CSV: JSON files allow for more contextual information to be sores while csv files are comma separated much simpler tabular structures.

To use JSON file in your program you must begin with importing JSON from Python library

```
import json

FILE_NAME: str = "Enrollments.json"  # initializing json file name as a constant

file = open(FILE_NAME, "w")   # to open a JSON file in Write mode
json.dump(students, file) # to read data from JSON file; requires much less code as compared to
```

csv
file.close() # close a JSON file

Validation using isalpha() method in Python: In Python, the isalpha() method is used for string validation, specifically to check if all characters in a string are alphabetic (letters). It returns True if the string contains only letters (uppercase or lowercase) and False otherwise. Spaces, numbers, and symbols will cause it to return False.

```
if not student_first_name.isalpha(): #check if user input is alphabets
    raise ValueError("The first name should not contain numbers.")
student_last_name = input("Enter the student's last name: ")
if not student_last_name.isalpha(): #check if user input is alphabets
    raise ValueError("The last name should not contain numbers.")
```

Figure 1 isalpha() use to check if user input is not string or alphabetic

ValueError exception

ValueError is an exception that occurs when a function receives an invalid value for input in python. This method can be used for handling validations by raising it when input data doesn't meet certain rules. This helps ensure data is correct and prevents errors in your program.

Exception

These are general exceptions in Python. See below example for except Exceptions ….for how this is used to perform error handling for non- specific errors (like ValueError, FileNotFound etc.)

```
except ValueError as e: # to catch Value error i.e. if user input for first and last name is not alphabetic then it cann
    print(e)  # Prints valueerror message stored in e
    print("-- Technical Error Message -- ")
    print(e.__doc__)
    print(e.__str__())
except Exception as e: # general exception handling for any other types of errors
    print("Error: There was a problem with your entered data.")
    print("-- Technical Error Message -- ")
    print(e.__doc__)
    print(e.__str__())
```

GitHub: is a cloud-based platform used for storing, sharing, and collaborating on code. It is built on Git, a version control system that tracks changes to files over time. GitHub can store repositories of code, which can be public or private. Saving repos and documentation with public access allows developers to review code and collaborate. GitHub also makes code to be version controlled such that changes can be traced, or code can be reverted as needed to a previous version in case of errors or bugs. Saving code with public access is how open source code applications are made available for free to the people.

Summary: In this module we covered advanced Python concepts like dictionaries, JSON file handling, input validation, and error handling. We also learnt GitHub as a tool for code collaboration and version control. These skills are essential for writing robust, maintainable, and collaborative Python programs.