

Rachel Hostetler

05/14/2024

Foundations of Programming: Python

Assignment05

# Assignment05

## Introduction

This week focused on integrating the previous knowledge of lists, tuples, and data processing in our scripts, with dictionaries and error handling.

## Creating the Script

First I opened Visual Studio, then opened “Assignment05-Starter.py,” where I added my information to the change log at the top (Figure 1).

```
#
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   RRoot,1/1/2030,Created Script
#   Rachel Hostetler, 05/13/24, Assignment05
# -----
```

Figure 1 – Header.

Next, I defined and updated my data constants and variables in the starter file based on the information given in Assignment04, this included adding an empty list and an empty dictionary (Figure 2).

```

# 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
-----
'''

# Define the Data Constants
FILE_NAME: str = "Enrollments.csv"

# Define the Data Variables and constants
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.
csv_data: str = '' # Holds combined string data separated by a comma.
file = None # Holds a reference to an opened file.
menu_choice: str # Hold the choice made by the user.
student_data: dict = {}
students: list = [] # a table of student data

```

Figure 2 – Constants and variables.

The next step included writing script where the CSV file was opened and then transformed into a dictionary. This section also includes script for handling errors, such as when a file cannot be found (Figure 3).

```

# When the program starts, read the file data into a list of lists (table)
# Extract the data from the file
try:
    file = open(FILE_NAME, "r")
    for row in file.readlines():
        # Transform the data to dictionary
        row_data = row.split(',')
        student_data = {"FirstName": row_data[0], "LastName": row_data[1], "Class": row_data[2].strip()}
        # Load it into our collection (list of lists)
        students.append(student_data)
except FileNotFoundError as e:
    print("File not found\n")
    print("-- More Info: Error -- ")
    print(e, e.__doc__, type(e), sep="\n")
except Exception as e:
    print("Other Error\n")
    print("-- More Info: Error -- ")
    print(e, e.__doc__, type(e), sep="\n")
finally:
    if not file.closed:
        file.close()

```

Figure 3 – Reading enrollments CSV file and transforming, while adding error handling.

The next step was to write script to present the Menu options listed in the data constants, and have a user input their first and last name and course name after selecting option 1. This section also includes error handling for the user inputs, such as displaying an error

message when numerical values are used in the first or last names. The last part of the script saves the user-defined information in a dictionary, and then prints a confirmation message saying that the student has been registered for the course (Figure 4).

```
# Present and Process the data
while (True):

    # Present the menu of choices
    print(MENU)
    menu_choice = input("What would you like to do: ")

    # 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: ")
        if not student_first_name.isalpha():
            raise ValueError("First name should not include numbers.")

        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("Last name should not include numbers.")

        course_name = input("Please enter the name of the course: ")
        student_data = {"FirstName": student_first_name, "LastName": student_last_name, "Class": course_name}
        students.append(student_data)
        print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
        continue
```

Figure 4 – Menu options and results for selecting option 1.

The next step is option 2, where I wrote script using a fstring that displays a message saying that all of the students in the list have been registered for their courses (Figure 5).

```
# 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['FirstName']} {student['LastName']} is enrolled in {student['Class']}")
    print("-"*50)
    continue
```

Figure 5. Menu option 2.

For menu option 3, I wrote script that saves this information to a CSV file and prints a confirmation message using the list after the saving is done. I also wrote script for error handling for when the file cannot be found, and an error message is then displayed (Figure 6).

```

# Save the data to a file
elif menu_choice == "3":

    try:
        file = open(FILE_NAME, "w")
        for student in students:
            csv_data = f"{student['FirstName']}, {student['LastName']}, {student['Class']}\n"
            file.write(csv_data)
        file.close()
        print("Data was saved to file!")
        for student in students:
            print(f"Student {student['FirstName']} {student['LastName']} is enrolled in {student['Class']}")

    except FileNotFoundError as e:
        print("File not found\n")
        print("-- More info: error -- ")
        print(e, e.__doc__, type(e), sep="\n")
    except Exception as e:
        print("Other error\n")
        print("-- More info: error -- ")
        print(e, e.__doc__, type(e), sep="\n")
    finally:
        if not file.closed:
            file.close()
    continue

```

Figure 6 – Menu option 3.

For menu option 4, I wrote script that ends the while loop and closes the program (Figure 7).

```

        file.close()
    continue

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

print("Program Ended")

```

Figure 7 – Menu option 4.

## Summary

In this assignment, I created a script which uses both user-defined variables and non user-defined variables, loops, lists, and dictionaries to present a menu of registration options for the user to interact with. Menu option 1 takes the students' first and last names and course name and saves these in a dictionary, option 2 presents all of the registered students

information, while option 3 saves this information to a CSV file and presents a confirmation message for registration. Both options 1 and 3 provide error messages when incorrect data (numerical) is input by the user, and when a file cannot be found.