Surafel Tebeje

05/24/2025

Foundations Of Programming: Python

Assignment05

GitHub: https://github.com/Surafel-01/IntroToProg-Python

Data processing using Dictionaries and file

Introduction:

In Python, a dictionary is a built-in data type that stores data in key-value pairs. It is unordered, mutable, and allows for fast access to values using unique keys. Dictionaries are defined using curly braces {}, with each key separated from its value by a colon, like {"name": "Alice", "age": 30}. Keys must be immutable types (like strings or numbers), while values can be any data type. Dictionaries are commonly used for structured data and fast lookups (ChatGPT). In this assignment paper, I will demonstrate how dictionaries and file can be used to process data.

Assignment instructions

Acceptance Criteria

Your program must include the following features and code to be accepted as complete:

File Name:

- The file is named Assignment05.py

Script Header:

1. The script header includes this text and has been updated with your name and the current date.

Constants:

- The constant **MENU: str** is set to the value:

```
---- 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
-----------------------------------------
```

- The constant **FILE_NAME: str** is set to the value "Enrollments.json"

- The constant values do not change throughout the program.

Variables:

- **student_first_name: str** is set to empty string.
- **student_last_name: str** is set to empty string.
- **course_name: str** is set to empty string.
- **file** is set to None.
- **menu_choice: str** is set to empty string.
- **student_data: dict** is set to an empty dictionary (This is changed from a list using in assignment04)
- **students: list:** list is set to and empty list

Input / Output:

1. On menu choice 1, the program prompts the user to enter the student's first name and last name, followed by the course name, using the input() function and stores the inputs in the respective variables.

2. Data collected for menu choice 1 is added to a dictionary named student_data. Next, student_data is added to the **students** two-dimensional list of dictionaries rows.

- On menu choice 2, the presents a string by formatting the collected data using the print() function.
- On menu choice 2, the program uses the print() function to show a string of comma-separated values for each row collected in the **students** variable.

Processing

- When the program starts, the contents of the "Enrollments.json" are automatically read into the **students** two-dimensional list of dictionary rows using the json.load() function. (**Tip:** Make sure to put some starting data into the file or you will get an error!)
- On menu choice 3, the program opens a file named "Enrollments.json" in write mode using the open() function. It writes the contents of the **students** variable to the file using the json.dump() function. Next, the file is closed using the close() method. Finally, the program displays what was written to the file using the **students** variable.
- On menu choice 4, the program ends.

Error Handling

- The program provides structured error handling when the file is read into the list of dictionary rows.
- The program provides structured error handling when the user enters a first name.
- The program provides structured error handling when the user enters a last name.
- The program provides structured error handling when the dictionary rows are written to the file.

Test:

- The program takes the user's input for a student's first, last name, and course name.
- The program displays the user's input for a student's first, last name, and course name.
- The program saves the user's input for a student's first, last name, and course name to a JSON file. (check this in a PyCharm or a simple text editor like Notepad or TextEdit.)
- The program allows users to enter multiple registrations (first name, last name, course name).

- The program allows users to display multiple registrations (first name, last name, course name).
- The program allows users to save multiple registrations to a file (first name, last name, course name).
- The program runs correctly in both **PyCharm and** from the **console or terminal**.

Source Control:

- The script file and the knowledge document are hosted on a GitHub repository.
- A link to the repository is included in the knowledge document.
- A link to the repository is included in the GitHub links forum.

The following syntax code was used in PyCharm and Console:

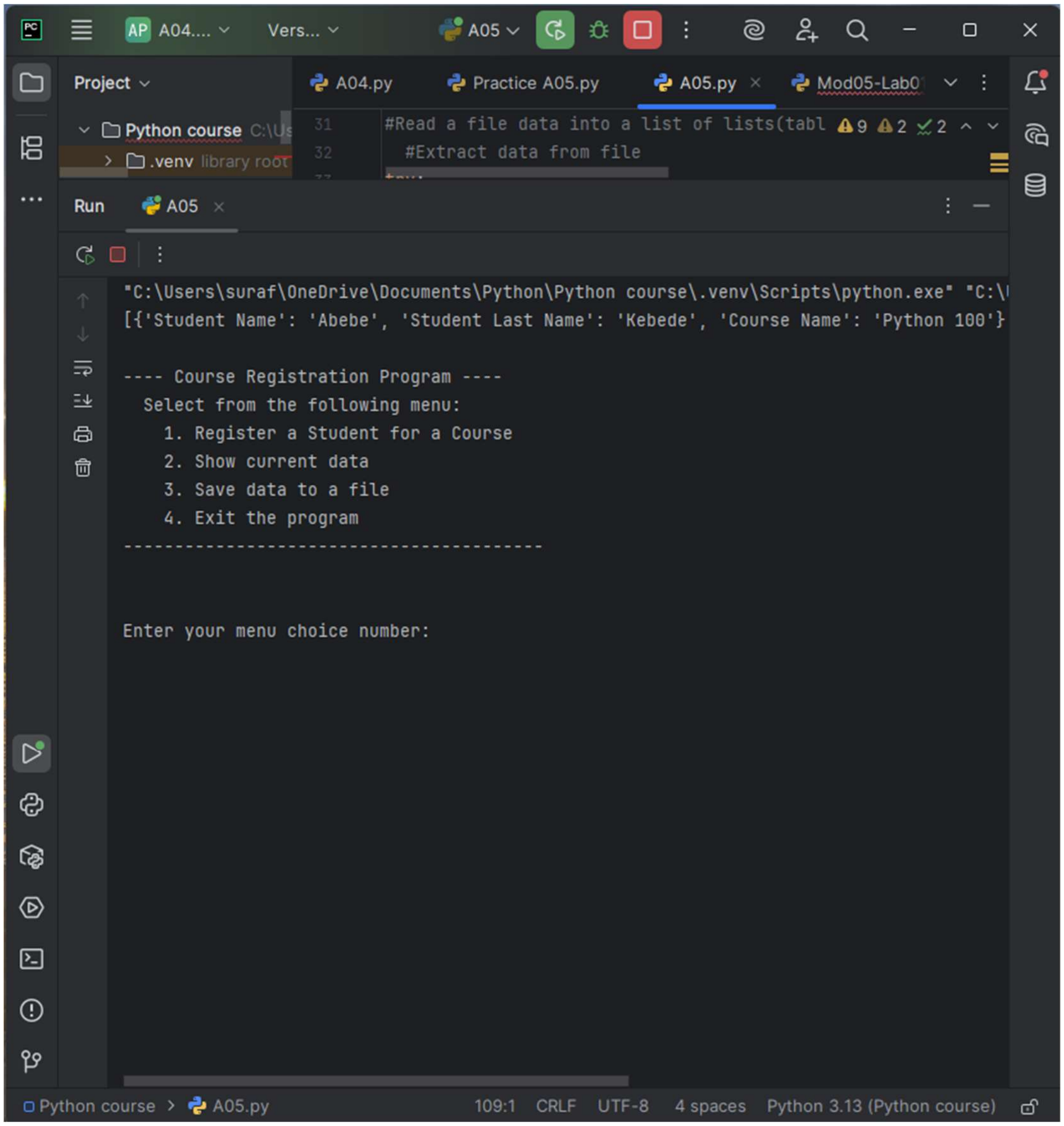
```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment04
# Desc: This assignment demonstrates using dictionaries and files to work with data
# Change Log: (Who, When, What)
#   Surafel Tebeje,05/24/2025,Created Script
# ------------------------------------------------------------------------------------------ #
# Define the Data Constants
import json

MENU:str='''\n---- 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
-----------------------------------------
\n'''

FILE_NAME:str="Enrollments.json"

#Define the data variables
student_first_name:str=''
student_last_name:str=''
course_name:str=''
file:object=None
menu_choice:str
student_data:dict={}
students:list=[]


#Read a file data into a list of lists(table)
  #Extract data from file
try:
    file=open(FILE_NAME,'r')    #This is thinking that there is already a csv file created, otherwise, I got
error message.
```

```
    students=json.load(file)
    file.close()
    print(students)
except FileNotFoundError as e:
    print("Text file must exist before running this script!\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()

#Display MENU
while True:
    print(MENU)
    menu_choice=input("Enter your menu choice number: ")
#Process data and custom messages
    #Colelct user input data
    if menu_choice=="1":
     try:
       student_first_name=input("What is the student's first name? ")
       if not student_first_name.isalpha():
           raise ValueError("The first name should not contain numbers")
       student_last_name=input("What is the student's last name? ")
       if not student_last_name.isalpha():
           raise ValueError("The last name should not contain numbers")
       course_name=input("What is the course's name? ")
       student_data={"Student Name":student_first_name, "Student Last
Name":student_last_name,"Course Name":course_name}
       students.append(student_data)
       print(f"\nstudent {student_first_name} {student_last_name} is registered!\n")
     except ValueError as e:
         print(e)
         print("-- Technical Error Message -- ")
         print(e.__doc__)
         print(e.str__())
     except Exception as e:
         print ("Error: There was a problem with your data!\n")
         print("---Technical Error Message--")
         print(e.__doc__)
         print(e.__str__())
         continue
     # Present the current data
    elif menu_choice == "2":
       for each in students:
           print(f"{each["Student Name"]},{each["Student Last Name"]},{each["Course Name"]}\n")
       continue
       # Save the data to a file
    elif menu_choice == "3":
       try:
           file = open(FILE_NAME, "w")
           json.dump(students,file)
           file.close()
           print("The following data is saved in the file:\n")
           print(students)
       except Exception as e:
           if file.closed== False:
             file.close()
           print("error: There was a problem in writing the file")
           print ("Please check that the file is not pen by another program")
           print("-- Technical Error Message -- ")
           print(e, e.__doc__)
           print(e.__str__())
       continue

    elif menu_choice=="4":
           print("program Ended!")
        #Stop the loop
           break
    else:
           print("Invalid menu choice")
```

Results from running the syntax codes: PyCharm

1. Reading file:

When the program reads the existing file, it reads the following data:

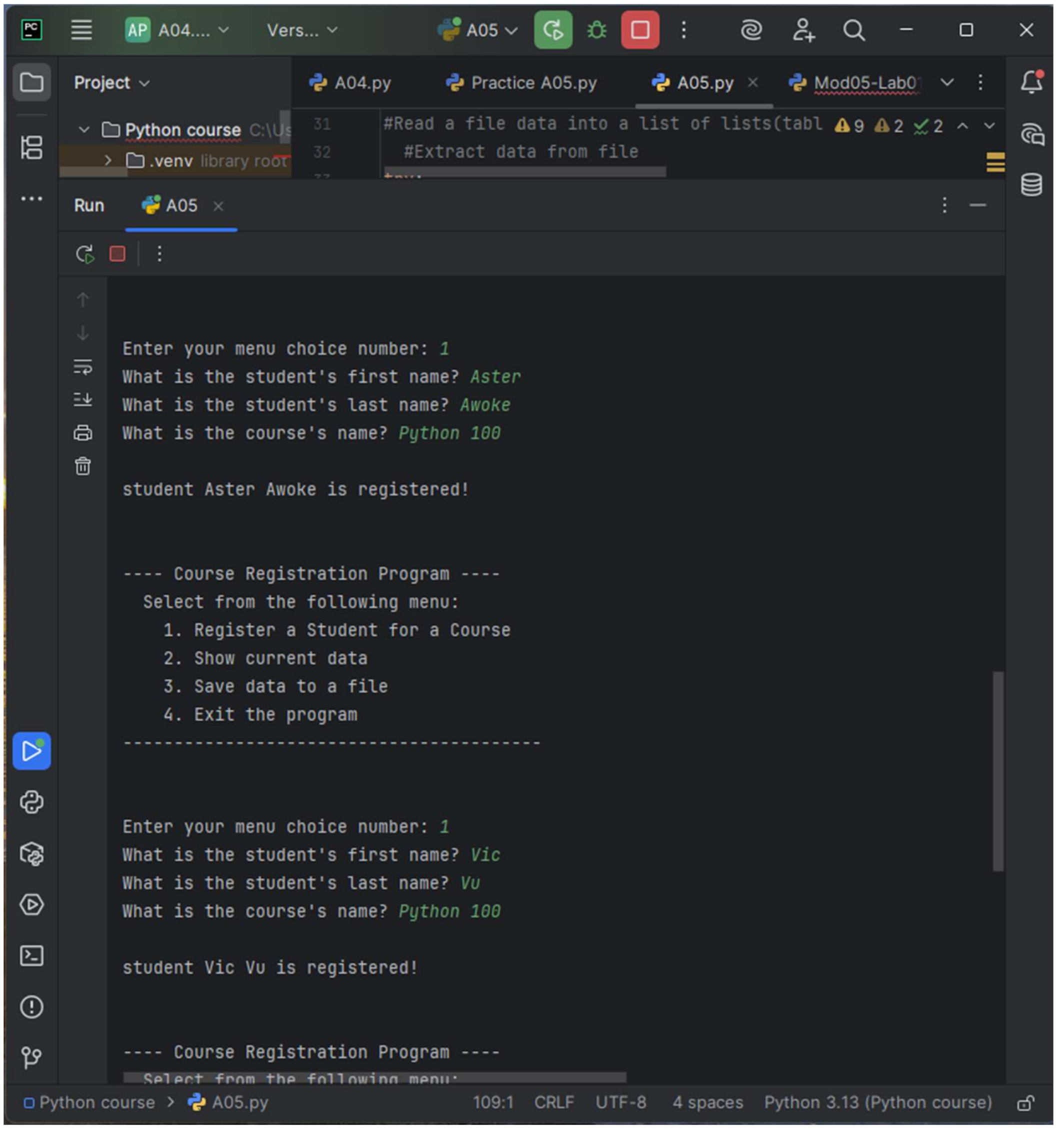


```
#Read a file data into a list of lists(tabl
  #Extract data from file
```

```
"C:\Users\suraf\OneDrive\Documents\Python\Python course\.venv\Scripts\python.exe" "C:\
[{'Student Name': 'Abebe', 'Student Last Name': 'Kebede', 'Course Name': '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
-----------------------------------------


Enter your menu choice number:
```

Figure 1: Results of reading the existing file

2. Results of Choice 1:

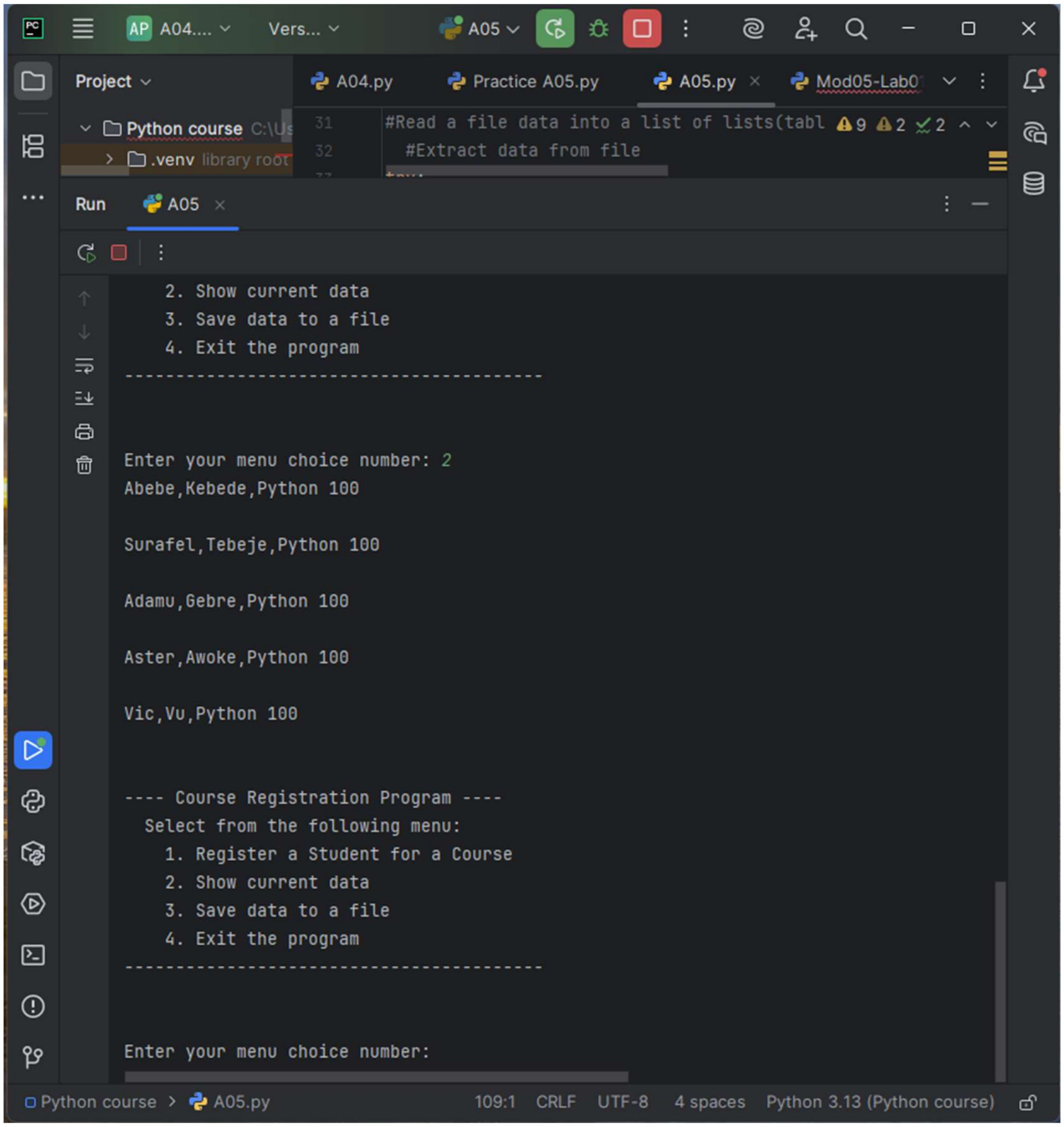


```
#Read a file data into a list of lists(tabl
  #Extract data from file
```

```
Enter your menu choice number: 1
What is the student's first name? Aster
What is the student's last name? Awoke
What is the course's name? Python 100

student Aster Awoke is registered!


---- 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
-----------------------------------------


Enter your menu choice number: 1
What is the student's first name? Vic
What is the student's last name? Vu
What is the course's name? Python 100

student Vic Vu is registered!


---- Course Registration Program ----
  Select from the following menu:
```

Figure 2: Results of choice 1

3. Results of Choice 2:

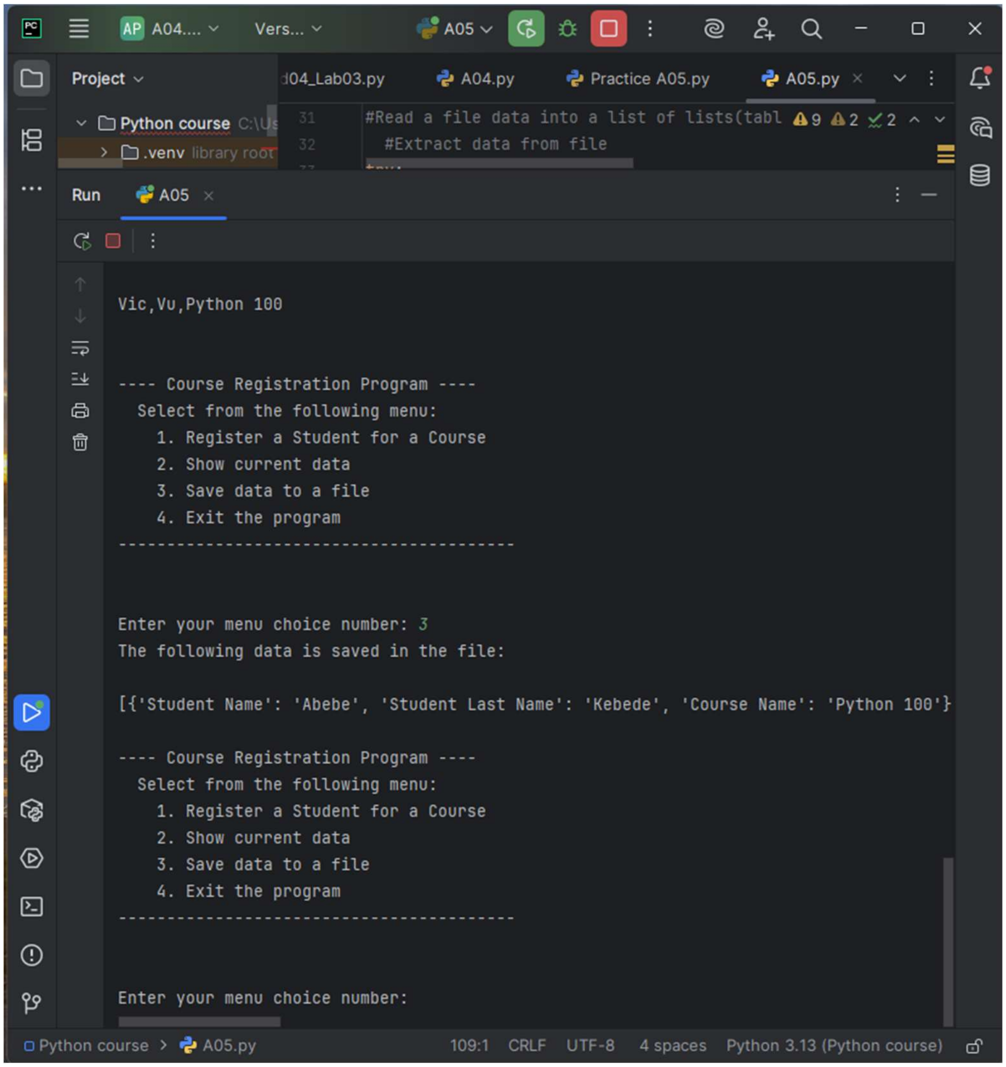


```
#Read a file data into a list of lists(tabl
  #Extract data from file
```

```
    2. Show current data
    3. Save data to a file
    4. Exit the program
-----------------------------------------



Enter your menu choice number: 2
Abebe,Kebede,Python 100

Surafel,Tebeje,Python 100

Adamu,Gebre,Python 100

Aster,Awoke,Python 100

Vic,Vu,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
-----------------------------------------



Enter your menu choice number:
```

Figure 3 : Results of choice 2

4. Results of Choice 3:

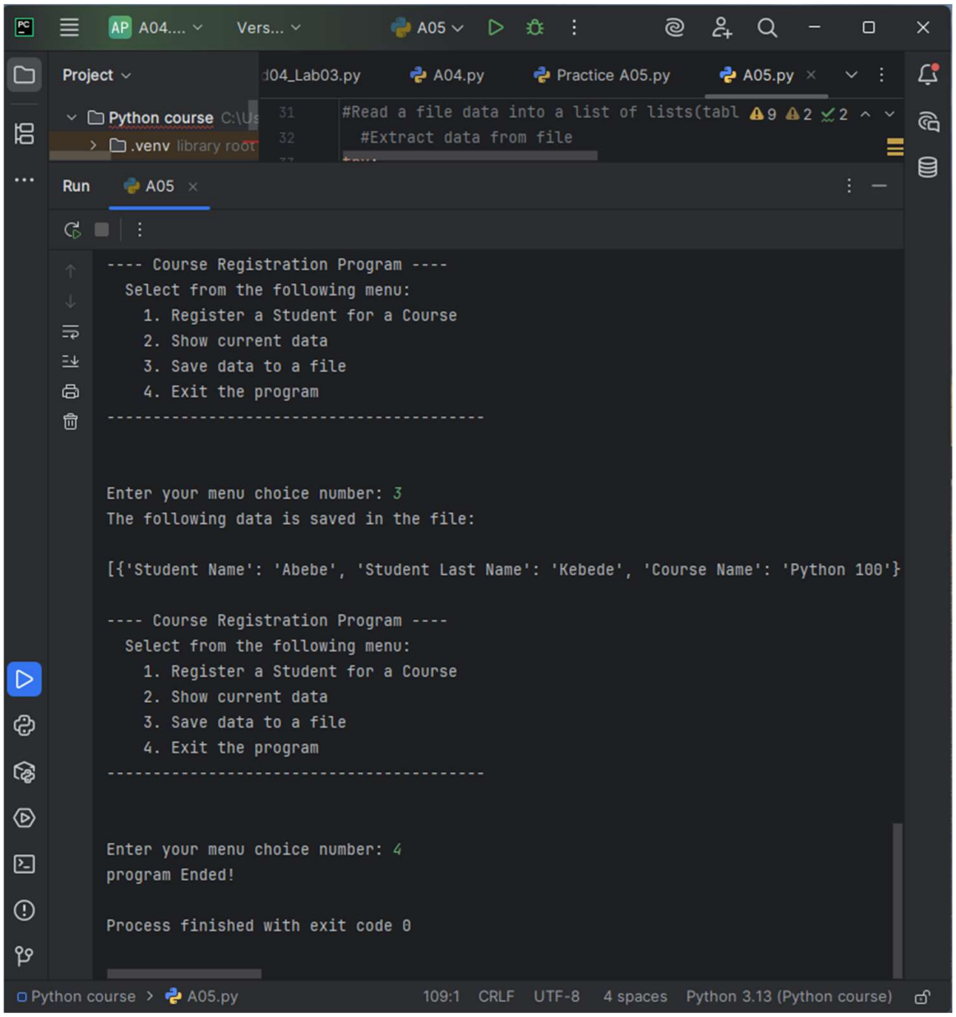


```
#Read a file data into a list of lists(tabl
  #Extract data from file
```

```
Vic,Vu,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
-----------------------------------------


Enter your menu choice number: 3
The following data is saved in the file:

[{'Student Name': 'Abebe', 'Student Last Name': 'Kebede', 'Course Name': '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
-----------------------------------------


Enter your menu choice number:
```

Figure 4: Results of choice 3

5. Results of Choice 4:

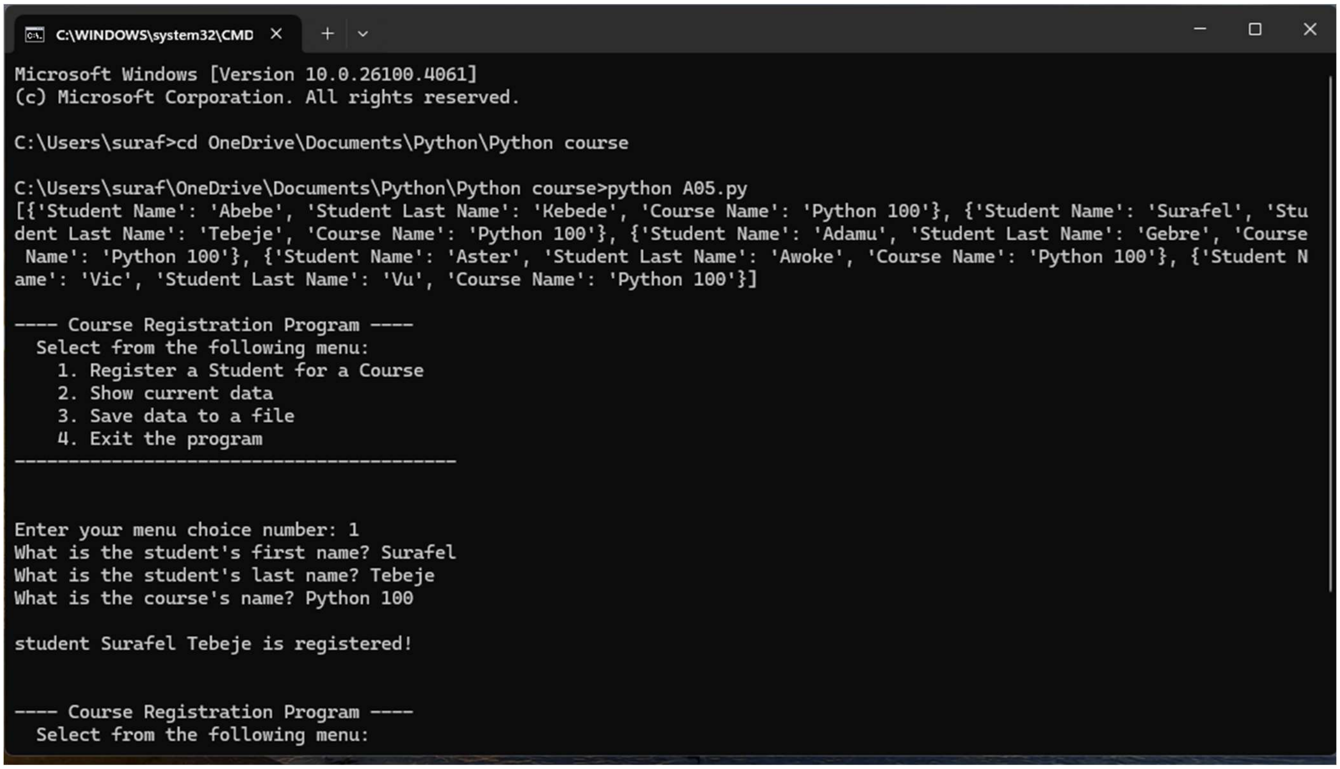


```
#Read a file data into a list of lists(tabl
  #Extract data from file
```

```
---- 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
-----------------------------------------


Enter your menu choice number: 3
The following data is saved in the file:

[{'Student Name': 'Abebe', 'Student Last Name': 'Kebede', 'Course Name': '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
-----------------------------------------


Enter your menu choice number: 4
program Ended!

Process finished with exit code 0
```

Figure 5: Results of choice 4

Running code on the console:



```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\suraf>cd OneDrive\Documents\Python\Python course

C:\Users\suraf\OneDrive\Documents\Python\Python course>python A05.py
[{'Student Name': 'Abebe', 'Student Last Name': 'Kebede', 'Course Name': 'Python 100'}, {'Student Name': 'Surafel', 'Stu
dent Last Name': 'Tebeje', 'Course Name': 'Python 100'}, {'Student Name': 'Adamu', 'Student Last Name': 'Gebre', 'Course
 Name': 'Python 100'}, {'Student Name': 'Aster', 'Student Last Name': 'Awoke', 'Course Name': 'Python 100'}, {'Student N
ame': 'Vic', 'Student Last Name': 'Vu', 'Course Name': '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
-----------------------------------------


Enter your menu choice number: 1
What is the student's first name? Surafel
What is the student's last name? Tebeje
What is the course's name? Python 100

student Surafel Tebeje is registered!


---- Course Registration Program ----
  Select from the following menu:
```

Figure 6: Results of the Console

Summary:

In summary, I demonstrated that dictionaries can be used to store tabular data with column heads as “keys” and “values” as variable values in dictionaries. I also demonstrated how to read a dictionary from a file, and how to open and write dictionaries into a file. I hope this paper gives a preliminary technical approach on how to use and extract dictionaries.