Nick Greco
Oct 31, 2024
Foundations of Programming: Python
Assignment04

Collections of Data within Python

Introduction

This assignment explains how to create a script for a basic student enrollment system using PyCharm IDE (Integrated Development Environment). The script builds on previous assignments by incorporating data collections to store and recall enrollment data. In Python, collections of data refer to built-in data structures that allow you to store, organize, and manipulate multiple items.

New Concepts Implemented

Data Collections

Data collections in Python provide developers with powerful tools for organizing, manipulating, and retrieving data. Here's an abbreviated list of what each type of collection can do:

Strings

- **Text Manipulation:** Strings allow for various operations like slicing, concatenation, searching, and formatting, essential for handling textual data.
- **Immutability:** Strings are immutable, which can prevent accidental changes and make string operations safe and predictable.

Tuples

- **Immutability:** Like strings, once created, tuples cannot be modified, making them useful for fixed collections of items.
- **Ordering:** Items are stored in a specific order, which can be accessed by index.
- **Memory Efficiency:** Tuples consume less memory compared to lists, making them more efficient for storing data that doesn't need to change.

Lists

- **Dynamic Sizing:** Lists can grow or shrink in size, allowing for flexible data management.

- **Ordering:** Items are stored in a specific order, which can be accessed by index.
- **Flexibility:** Lists support a wide range of methods for adding, modifying, and deleting elements.
- **Multi-Dimensional:** Lists can contain other lists or various data types, enabling complex data structures.

Sets

- **Uniqueness:** Sets automatically eliminate duplicate values, ensuring all items are unique.
- **Unordered:** Since sets are unordered, they provide a fast way to check for membership without worrying about order.

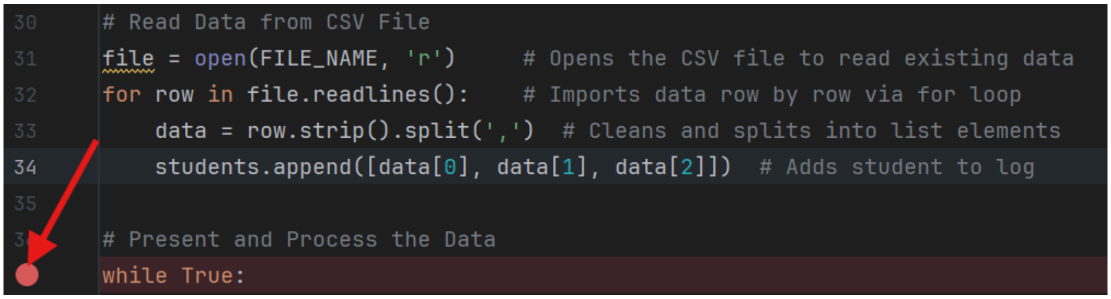
Dictionaries

- **Key-Value Pairing:** Dictionaries store data as key-value pairs, enabling efficient lookups, additions, and modifications based on unique keys.

Debugging in PyCharm IDE

Debugging is the process of identifying, analyzing, and fixing errors in the code, which can lead to incorrect results, crashes, or unintended behavior. Debugging is a critical part of the software development process and involves various techniques and tools to ensure the code runs as intended.

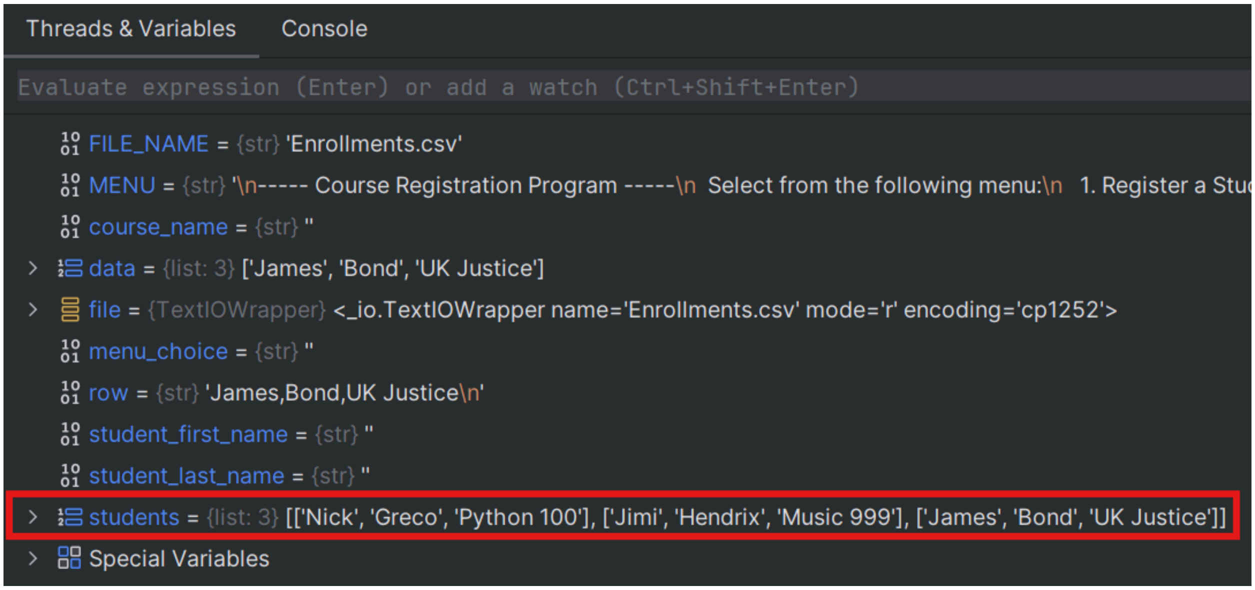
Breakpoints in PyCharm are a powerful debugging feature that allows developers to pause the execution of their code at a specific line. This enables them to inspect the state of the application, including variable values, the flow of control, and the overall behavior of the program at that point in time. Figure 1 illustrates how to set a breakpoint to evaluate the variable “students” in this assignment.



```
# Read Data from CSV File
file = open(FILE_NAME, 'r')      # Opens the CSV file to read existing data
for row in file.readlines():     # Imports data row by row via for loop
    data = row.strip().split(',')  # Cleans and splits into list elements
    students.append([data[0], data[1], data[2]])  # Adds student to log

# Present and Process the Data
while True:
```

Figure 1: Adding a Breakpoint in ‘Assignment04’

Variables can be examined for their content at a breakpoint in the code. In Figure 2 below, the content of variable “students” was examined to ensure that data had been properly imported from the CSV file and parsed prior to initiation of the while loop.


Threads & Variables
Console
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

```
FILE_NAME = {str} 'Enrollments.csv'
MENU = {str} '\n----- Course Registration Program -----\n  Select from the following menu:\n   1. Register a Stu
course_name = {str} ''
> data = {list: 3} ['James', 'Bond', 'UK Justice']
> file = {TextIOWrapper} <_io.TextIOWrapper name='Enrollments.csv' mode='r' encoding='cp1252'>
menu_choice = {str} ''
row = {str} 'James,Bond,UK Justice\n'
student_first_name = {str} ''
student_last_name = {str} ''
> students = {list: 3} [['Nick', 'Greco', 'Python 100'], ['Jimi', 'Hendrix', 'Music 999'], ['James', 'Bond', 'UK Justice']]
> Special Variables
```

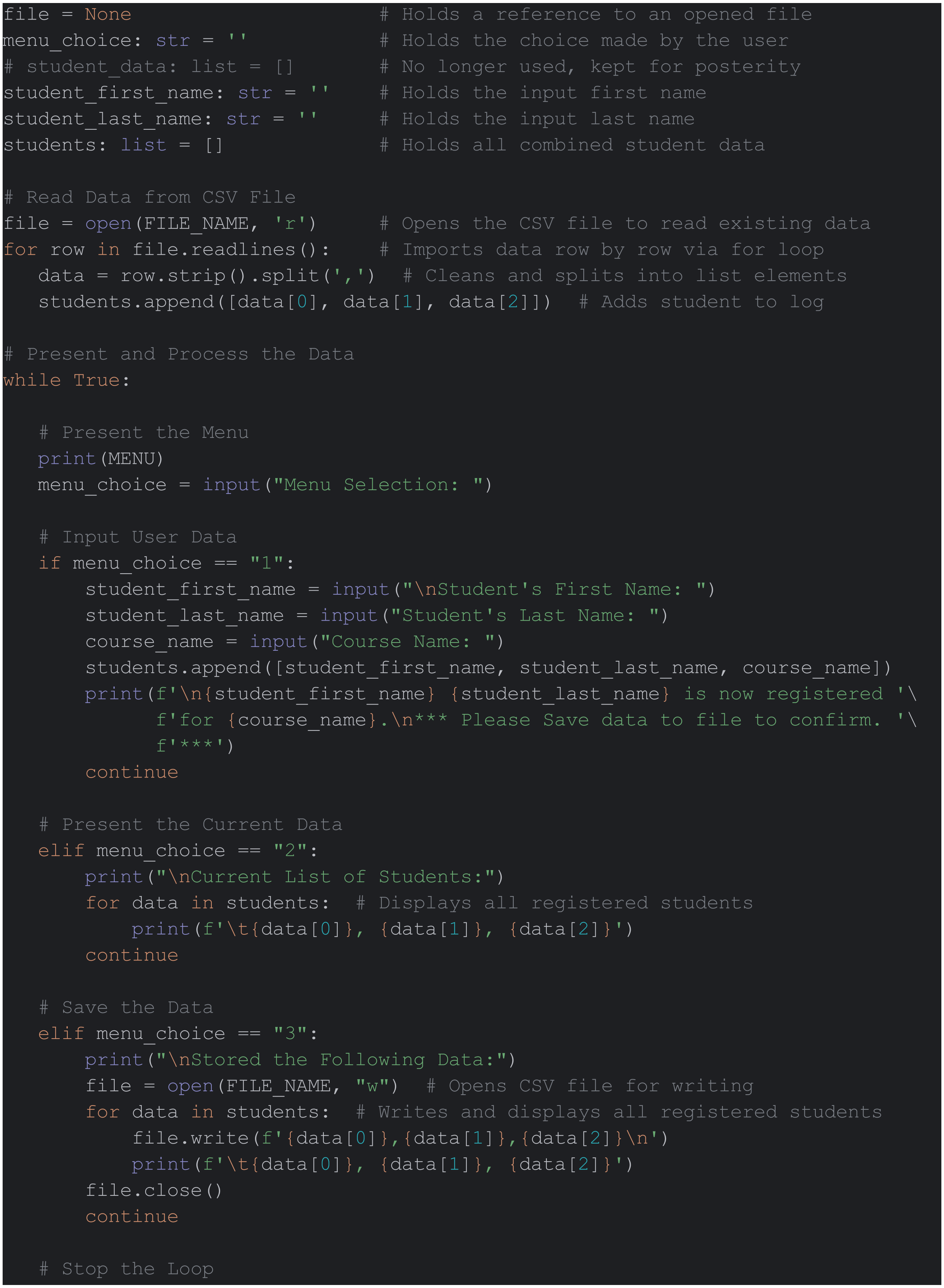
Figure 2: Threads and Variables at Breakpoint

Creating the Script

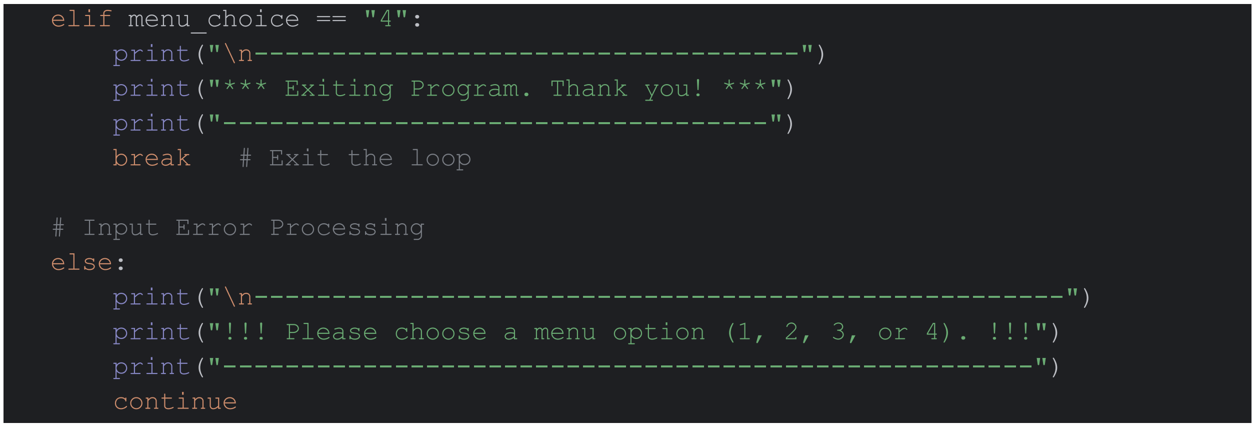
Overview

This script is a course registration program that allows users to register students, view current data, and save it to a CSV file. Figure 3 below shows the completed script for this assignment.

```
# -------------------------------------------------------------------------- #
# Title: Assignment04
# Desc:  This assignment demonstrates using lists and files to work with data
# Change Log: (Who, What, When)
#   N.Greco, 10/28/2024, Created Script
# -------------------------------------------------------------------------- #

# 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.csv"

# Define the Data Variables
course_name: str = ''           # Holds the input course name
# csv_data: str = ''            # No longer used, kept for posterity
```



```
file = None                     # Holds a reference to an opened file
menu_choice: str = ''           # Holds the choice made by the user
# student_data: list = []       # No longer used, kept for posterity
student_first_name: str = ''    # Holds the input first name
student_last_name: str = ''     # Holds the input last name
students: list = []             # Holds all combined student data

# Read Data from CSV File
file = open(FILE_NAME, 'r')     # Opens the CSV file to read existing data
for row in file.readlines():    # Imports data row by row via for loop
    data = row.strip().split(',')  # Cleans and splits into list elements
    students.append([data[0], data[1], data[2]])  # Adds student to log

# Present and Process the Data
while True:

    # Present the Menu
    print(MENU)
    menu_choice = input("Menu Selection: ")

    # Input User Data
    if menu_choice == "1":
        student_first_name = input("\nStudent's First Name: ")
        student_last_name = input("Student's Last Name: ")
        course_name = input("Course Name: ")
        students.append([student_first_name, student_last_name, course_name])
        print(f'\n{student_first_name} {student_last_name} is now registered '\
              f'for {course_name}.\n*** Please Save data to file to confirm. '\
              f'***')
        continue

    # Present the Current Data
    elif menu_choice == "2":
        print("\nCurrent List of Students:")
        for data in students:  # Displays all registered students
            print(f'\t{data[0]}, {data[1]}, {data[2]}')
        continue

    # Save the Data
    elif menu_choice == "3":
        print("\nStored the Following Data:")
        file = open(FILE_NAME, "w")  # Opens CSV file for writing
        for data in students:  # Writes and displays all registered students
            file.write(f'{data[0]},{data[1]},{data[2]}\n')
            print(f'\t{data[0]}, {data[1]}, {data[2]}')
        file.close()
        continue

    # Stop the Loop
```



```
    elif menu_choice == "4":
        print("\n-----------------------------------")
        print("*** Exiting Program. Thank you! ***")
        print("-----------------------------------")
        break   # Exit the loop

    # Input Error Processing
    else:
        print("\n----------------------------------------------------")
        print("!!! Please choose a menu option (1, 2, 3, or 4). !!!")
        print("----------------------------------------------------")
        continue
```

Figure 3: ‘Assignment04’ Python Script

Key Functionalities

1. **Constants and Variables:** The script defines a menu string (“MENU”) that presents four options: registering a student, showing current data, saving data to a file, and exiting the program. A constant “FILE_NAME” is defined to specify the CSV file (‘Enrollments.csv’) where student data will be stored. Several variables are initialized to hold user input and student data.
2. **Reading Data from CSV:** The script opens the specified CSV file to read existing student enrollment data. It processes each line in the file, cleans the data, splits it into components, and appends each student’s data (first name, last name, course name) to a list called “students”.
3. **Main Program Loop:** A while loop runs indefinitely, presenting the user with the menu options until they choose to exit.
4. **Menu Options:**
    - **Register a Student**: Prompts the user for the student's first name, last name, and course name. Appends the new student's data to the “students” list and informs the user to save the data to confirm the registration.
    - **Show Current Data**: Displays the current list of registered students by iterating over the students list and printing each student's details.
    - **Save Data to a File**: Opens the CSV file in write mode, writes all registered students' data back to the file, and displays the saved data to the user.
    - **Exit the Program**: Breaks the loop, effectively ending the program, and thanks the user for using the application.
5. **Input Validation:** If the user enters an invalid menu option (anything other than 1, 2, 3, or 4), the program informs them and prompts them to make a valid selection again.

Concepts Demonstrated

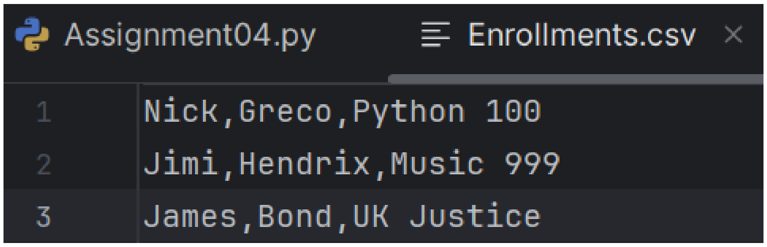
The script demonstrates several concepts from previous assignments, along with the following additions:

1. **Data Structures:** The script utilizes lists to store student data. The students list holds sublists, each representing a student with their first name, last name, and course name. This demonstrates how lists can be used to store collections of related data.
2. **File Handling:** The script opens a CSV file (‘Enrollments.csv’) to read existing student data. It demonstrates how to read data from a file line by line and parse it into a usable format (splitting strings into lists). The script also allows for saving the student data back to the CSV file. It demonstrates how to open a file for writing, format data, and save it to a file.
3. **Control Flow:** The use of a while True loop allows the program to continuously present the menu until the user decides to exit (menu choice "4"). This demonstrates the concept of loops for repeated execution. The if-elif-else structure processes the user's menu selection. It checks the user's input and executes different blocks of code based on the choice, demonstrating control flow through conditional logic.

Running the Script

Using PyCharm IDE

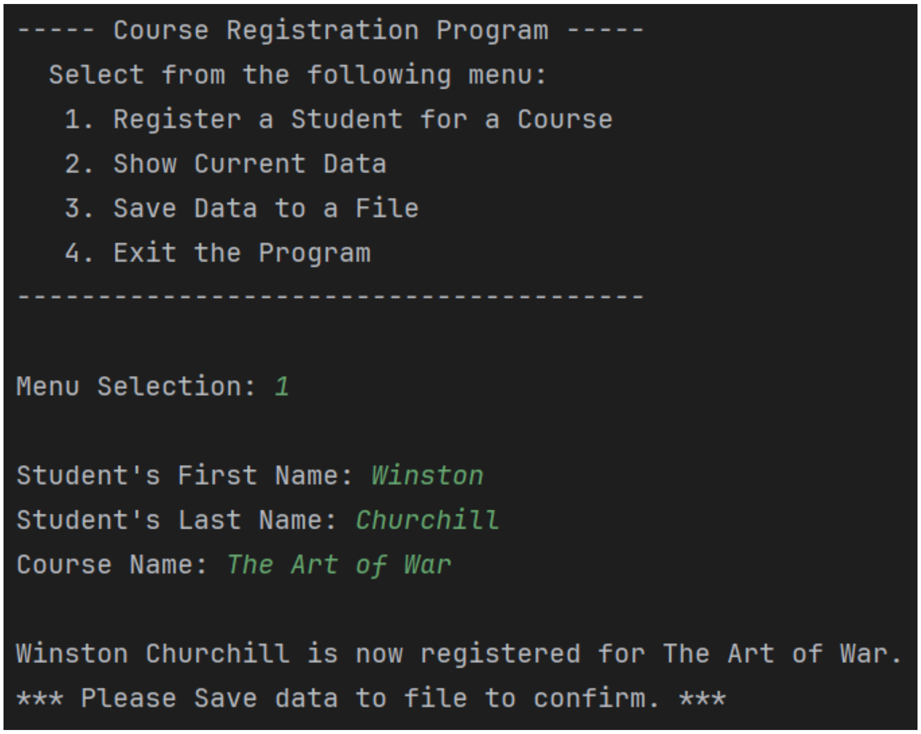
Figure 4 shows the contents of the ‘Enrollments.csv’ file prior to running the ‘Assignment04’ script. Data was manually entered to enable the program to run properly (per assignment instructions).



```
Nick,Greco,Python 100
Jimi,Hendrix,Music 999
James,Bond,UK Justice
```

Figure 4: ‘Enrollments.csv’ Data Contents before running Script

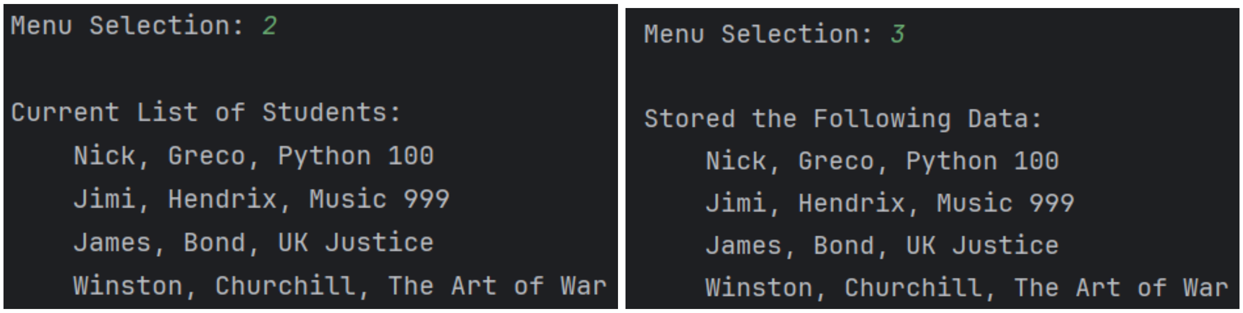
The script was initially run using PyCharm. Figure 5 shows the menu presented to the user and displayed prompts when selection “1” is chosen. Example user input is shown in green text.



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


Menu Selection: 1

Student's First Name: Winston
Student's Last Name: Churchill
Course Name: The Art of War

Winston Churchill is now registered for The Art of War.
*** Please Save data to file to confirm. ***
```

Figure 5: Menu and Option "1" Prompts

Figure 6 shows the program outputs when menu options 2 (left) and 3 (right) are chosen.

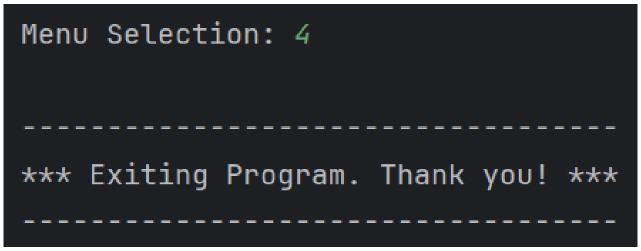


```
Menu Selection: 2

Current List of Students:
    Nick, Greco, Python 100
    Jimi, Hendrix, Music 999
    James, Bond, UK Justice
    Winston, Churchill, The Art of War
```

```
Menu Selection: 3

Stored the Following Data:
    Nick, Greco, Python 100
    Jimi, Hendrix, Music 999
    James, Bond, UK Justice
    Winston, Churchill, The Art of War
```

Figure 6: Menu Options "2" (left) and "3" (right) Outputs

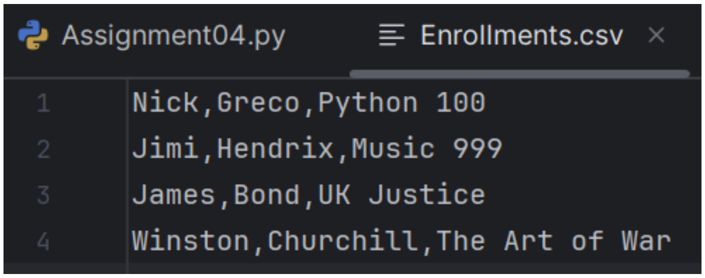
Figure 7 shows the program output when menu option 4 is chosen.



```
Menu Selection: 4


-----------------------------------
*** Exiting Program. Thank you! ***
-----------------------------------
```

Figure 7: Menu Option "4" Output

Figure 8 shows the contents of the 'Enrollments.csv' file after running the 'Assignment04' script and providing user input to register a single additional student.

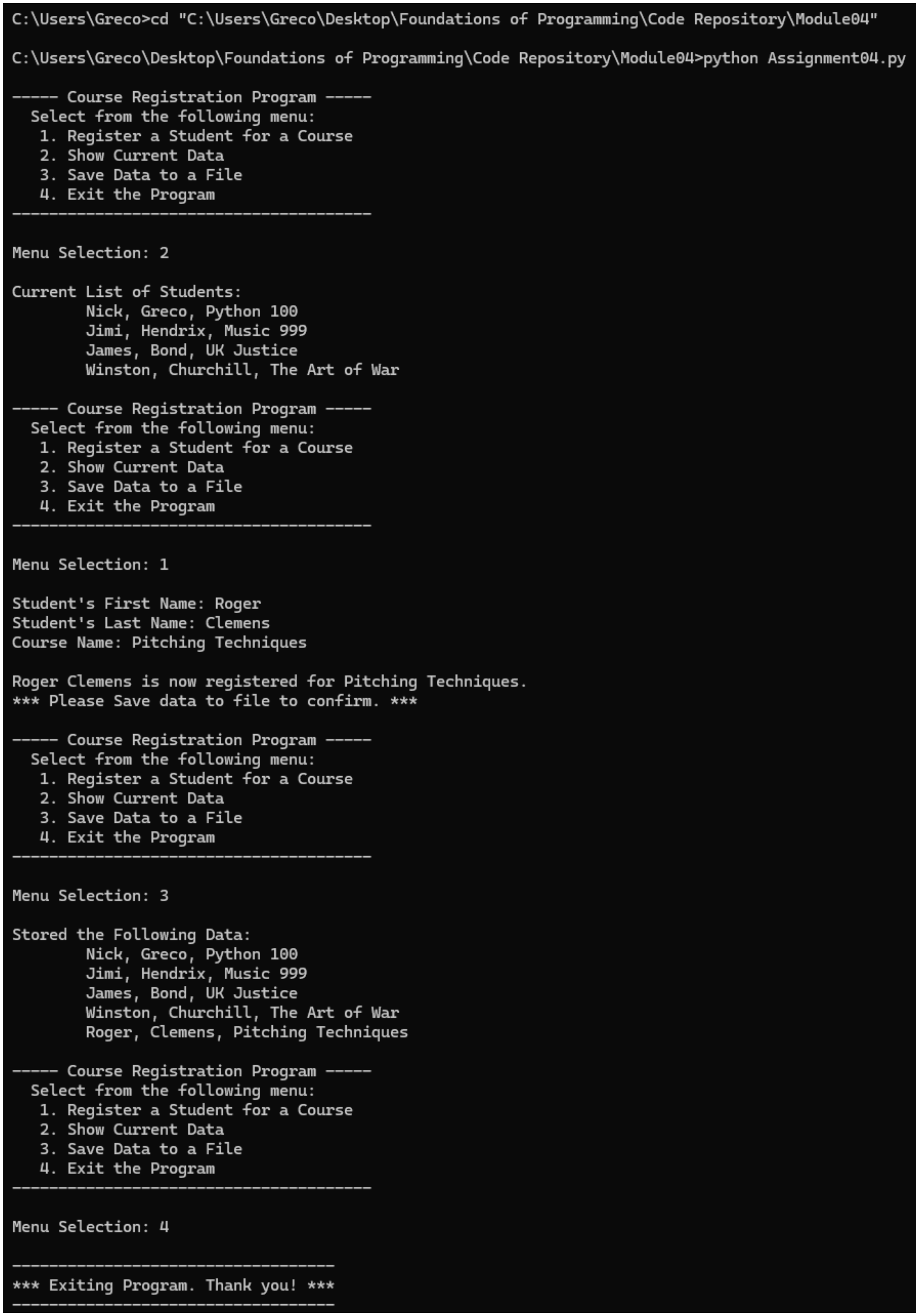


```
Nick,Greco,Python 100
Jimi,Hendrix,Music 999
James,Bond,UK Justice
Winston,Churchill,The Art of War
```

Figure 8: ‘Enrollments.csv’ Data Contents after running Script

Using Windows Command Prompt

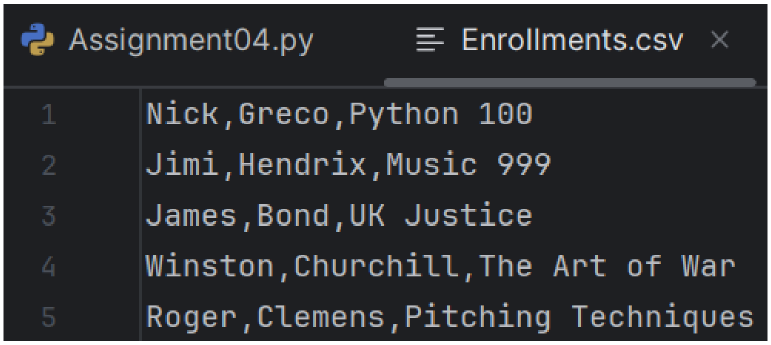
Figure 9 shows running the same script using Windows Command Prompt.



```
C:\Users\Greco>cd "C:\Users\Greco\Desktop\Foundations of Programming\Code Repository\Module04"

C:\Users\Greco\Desktop\Foundations of Programming\Code Repository\Module04>python Assignment04.py

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

Menu Selection: 2

Current List of Students:
        Nick, Greco, Python 100
        Jimi, Hendrix, Music 999
        James, Bond, UK Justice
        Winston, Churchill, The Art of War

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

Menu Selection: 1

Student's First Name: Roger
Student's Last Name: Clemens
Course Name: Pitching Techniques

Roger Clemens is now registered for Pitching Techniques.
*** Please Save data to file to confirm. ***

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

Menu Selection: 3

Stored the Following Data:
        Nick, Greco, Python 100
        Jimi, Hendrix, Music 999
        James, Bond, UK Justice
        Winston, Churchill, The Art of War
        Roger, Clemens, Pitching Techniques

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

Menu Selection: 4

-----------------------------------
*** Exiting Program. Thank you! ***
-----------------------------------
```

Figure 9: Using Windows Command Prompt to run 'Assignment04.py'

Figure 10 shows the contents of the CSV file after executing the script using Command Prompt.



```
Nick,Greco,Python 100
Jimi,Hendrix,Music 999
James,Bond,UK Justice
Winston,Churchill,The Art of War
Roger,Clemens,Pitching Techniques
```

Figure 10: CSV File Contents after running Script

Summary

This document outlines the creation of a basic student enrollment system using Python, highlighting the use of data collections for storing and managing enrollment data. It discusses various data structures—such as strings, tuples, lists, sets, and dictionaries—along with their functionalities and how they enhance data organization and manipulation. Additionally, this document covers debugging techniques in the PyCharm IDE, explains the script's main functionalities for user interaction, and emphasizes key programming concepts such as file handling, control flow, and input validation.

Citations

OpenAI ChatGPT. (October 2024). https://chatgpt.com/: Aspects of this assignment were informed and created by queries I submitted to ChatGPT.