Jesse Huminski

10/22/2024

IT FDN 100 A

Assignment 02

Saving Input Data To A File

Introduction

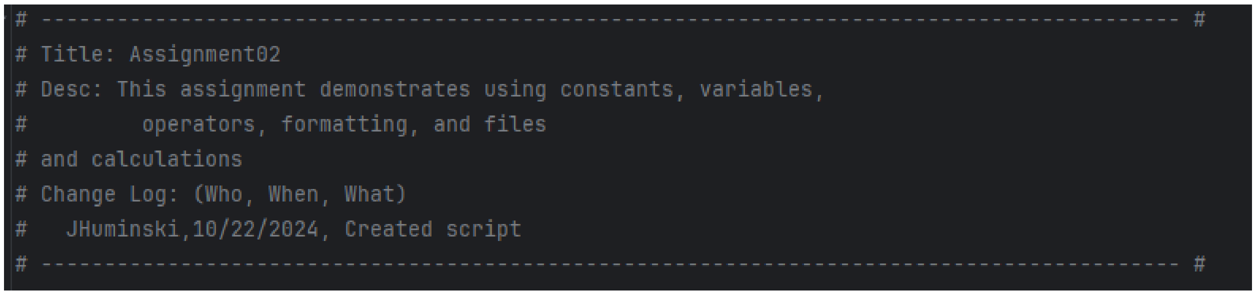
In module 02, we continued to learn how different data can be interpreted by the Python Language, we defined what a string is and went into further detail about string concatenation. Additionally, we covered the different types of errors we may come across as we are coding our program, including data type conversion errors, syntax, logic, and runtime errors, and compile type errors. We also discussed how to format data properly, how to output that formatted data into a file, and discussed the different operators we can use within the language.

Lastly, we went over the ways to troubleshoot and debug issues and how important it is to document those troubleshooting steps to help avoid problems in the future and to potentially help other programmers who may face a similar issue.

This week, we were tasked with creating a program that expands upon what we learned in Module 01 and requests that we use different methods of data formatting to be more concise and enhance our program's readability.

Creating the Program

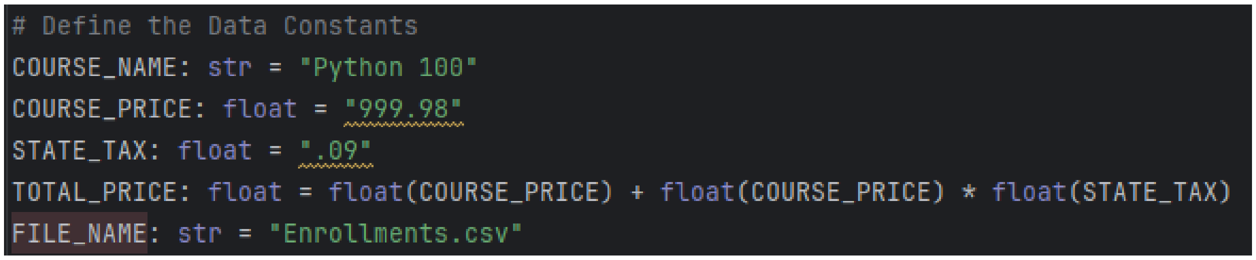
As always, we start with the program header to display the necessary information of the program. (Image 1.1)



```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment02
# Desc: This assignment demonstrates using constants, variables,
#         operators, formatting, and files
# and calculations
# Change Log: (Who, When, What)
#   JHuminski,10/22/2024, Created script
# ------------------------------------------------------------------------------------------ #
```

Image 1.1: The header of the script, "Assignment 02."

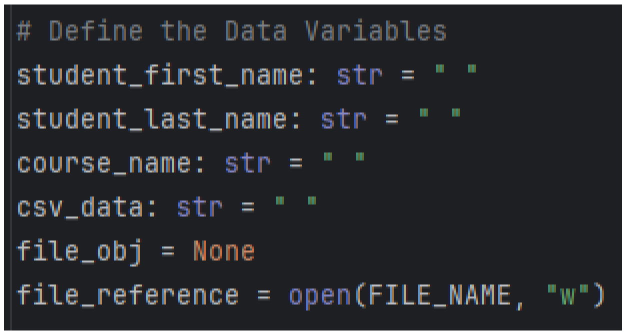
From there, I defined the data constants that I would be using in the script and ensured that all constants were spelled and data typed accurately. I also created a string concatenation that adds the STATE_TAX to the COURSE_PRICE and does the arithmetic properly. Lastly, I also defined the FILE_NAME that I would like the program to create and write to save the input information later in the program. (Image 1.2)



```
# Define the Data Constants
COURSE_NAME: str = "Python 100"
COURSE_PRICE: float = "999.98"
STATE_TAX: float = ".09"
TOTAL_PRICE: float = float(COURSE_PRICE) + float(COURSE_PRICE) * float(STATE_TAX)
FILE_NAME: str = "Enrollments.csv"
```

Image 1.2: The constants defined in the script.

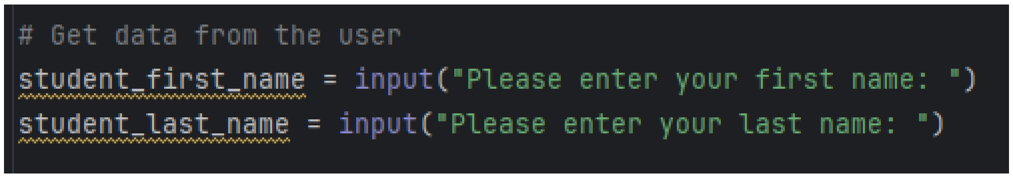
Next, I needed to define the data variables. Since I am going to be requesting a person's name, we utilized "student_first_name" and "student_last_name" as we did in Module 01. Here, we also see a new variable that utilizes the open() function. This allows the program to open and write data to an existing file or create a new file if the file is not already there. (Image 1.3)



```python
# Define the Data Variables
student_first_name: str = " "
student_last_name: str = " "
course_name: str = " "
csv_data: str = " "
file_obj = None
file_reference = open(FILE_NAME, "w")
```

Image 1.3: Defining the data variables.

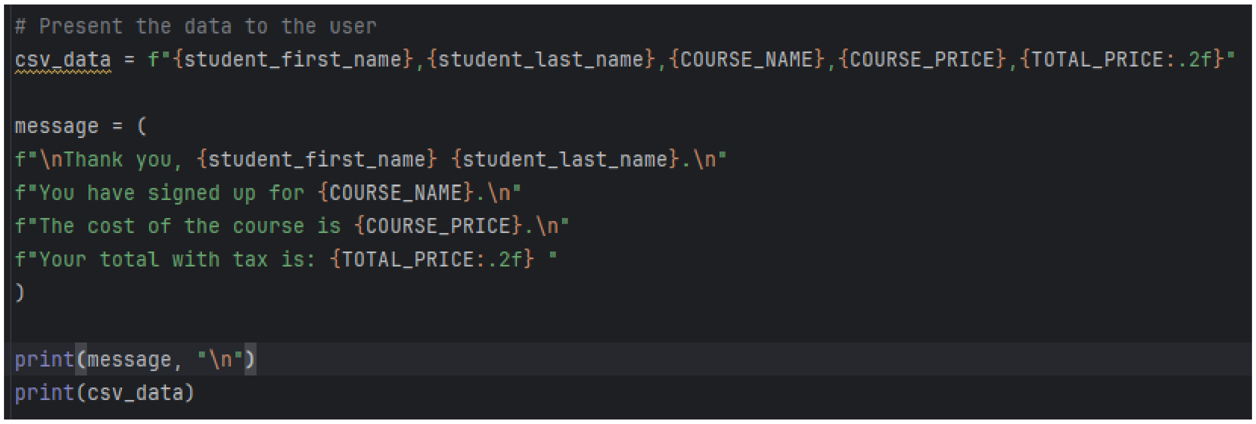
As it was in Module 01, we are being tasked with gathering information from the user to store into variables "student_first_name" and "student_last_name." Here, I used the input() function to ensure that the program is requesting information from the user appropriately. (Image 1.4)



```python
# Get data from the user
student_first_name = input("Please enter your first name: ")
student_last_name = input("Please enter your last name: ")
```

Image 1.4: Gathering information from the user.

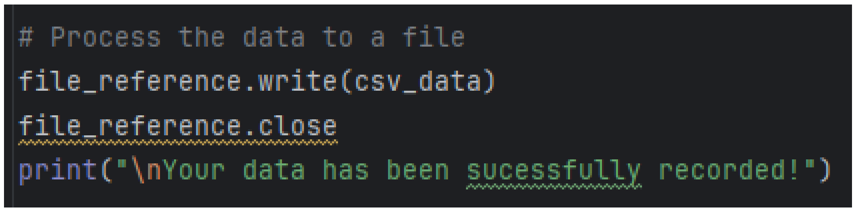
We are then tasked with presenting the data to the user. Initially, I defined the variable "csv_data" using an f-string for cleaner readability. I also formatted the "TOTAL_PRICE" constant to ensure that it only displayed two digits after the decimal point. From there, I created a message that informs the student that they have successfully signed up for the Python 100 course and included the "COURSE_PRICE" and "TOTAL_PRICE" constants. The information is then printed on the screen to the user. (Image 1.5)



```python
# Present the data to the user
csv_data = f"{student_first_name},{student_last_name},{COURSE_NAME},{COURSE_PRICE},{TOTAL_PRICE:.2f}"

message = (
f"\nThank you, {student_first_name} {student_last_name}.\n"
f"You have signed up for {COURSE_NAME}.\n"
f"The cost of the course is {COURSE_PRICE}.\n"
f"Your total with tax is: {TOTAL_PRICE:.2f} "
)

print(message, "\n")
print(csv_data)
```

Image 1.5: Displaying the data to the user

Lastly, I wrote the collected data to a CSV file using the constant "FILE_NAME." This generated a CSV file in the relative path. Once the data is written to the CSV file, a message information the user that data has successfully been captured is generated. (Image 1.6)

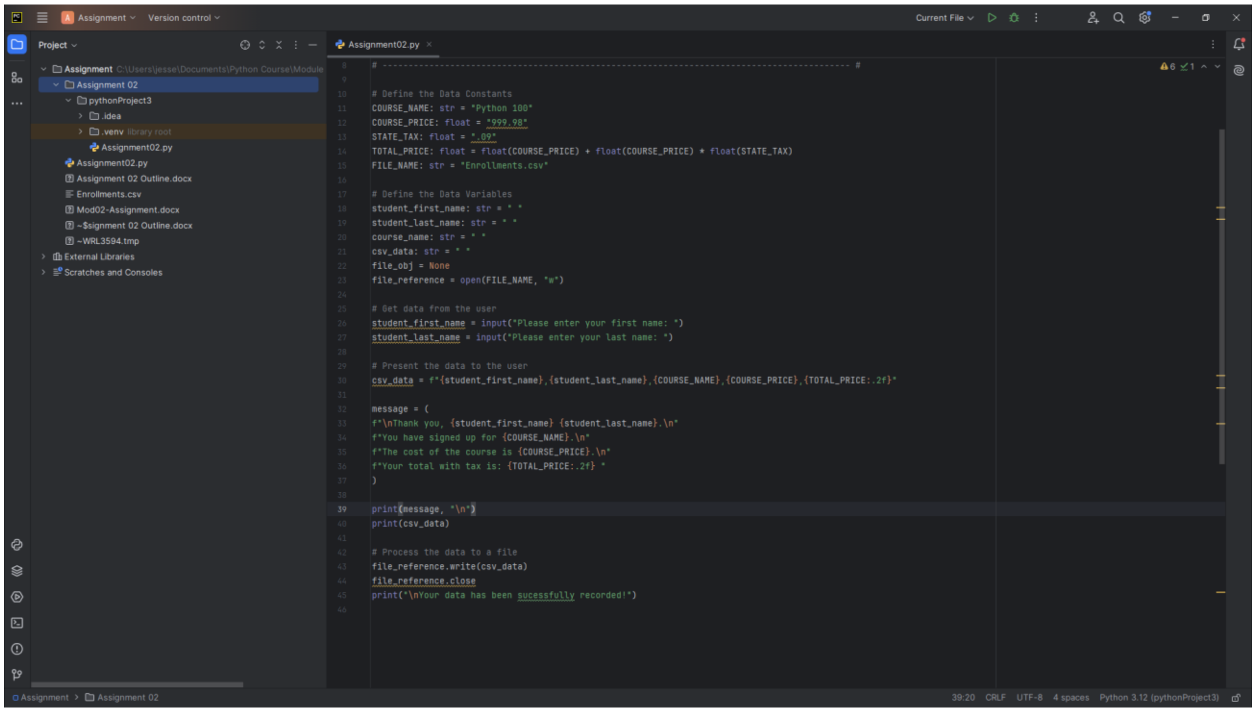


```python
# Process the data to a file
file_reference.write(csv_data)
file_reference.close
print("\nYour data has been sucessfully recorded!")
```

Image 1.6: Writing the data and creating the CSV file

Program Testing

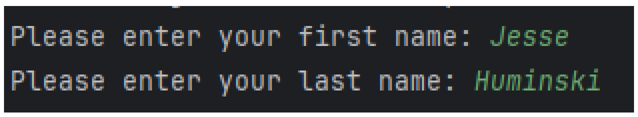
After compiling the requested code in PyCharm, it is now time to test the code to make sure that everything is working as expected. Since I am using PyCharm, I can run the code directly in this program. (Image 2.1)



```python
# ------------------------------------------------------------------------------------------ #

# Define the Data Constants
COURSE_NAME: str = "Python 100"
COURSE_PRICE: float = "999.98"
STATE_TAX: float = ".09"
TOTAL_PRICE: float = float(COURSE_PRICE) + float(COURSE_PRICE) * float(STATE_TAX)
FILE_NAME: str = "Enrollments.csv"

# Define the Data Variables
student_first_name: str = " "
student_last_name: str = " "
course_name: str = " "
csv_data: str = " "
file_obj = None
file_reference = open(FILE_NAME, "w")

# Get data from the user
student_first_name = input("Please enter your first name: ")
student_last_name = input("Please enter your last name: ")

# Present the data to the user
csv_data = f"{student_first_name},{student_last_name},{COURSE_NAME},{COURSE_PRICE},{TOTAL_PRICE:.2f}"

message = (
f"\nThank you, {student_first_name} {student_last_name}.\n"
f"You have signed up for {COURSE_NAME}.\n"
f"The cost of the course is {COURSE_PRICE}.\n"
f"Your total with tax is: {TOTAL_PRICE:.2f} "
)

print(message, "\n")
print(csv_data)

# Process the data to a file
file_reference.write(csv_data)
file_reference.close
print("\nYour data has been sucessfully recorded!")
```

Image 2.1: PyCharm displaying my generated code.

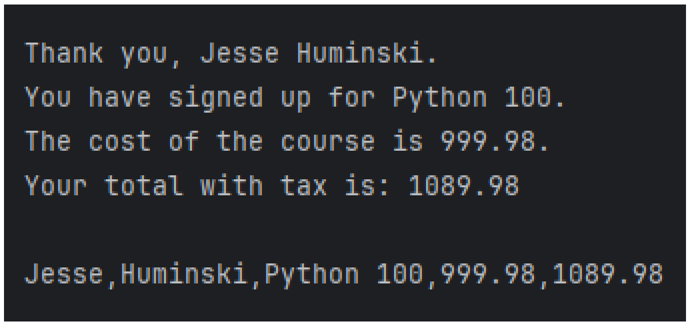
I run the code and am displayed a prompt requesting my first name. Once I enter the information and press enter, I am then prompted to enter my last name. So far, the code is working as expected. (Image 2.2)



```
Please enter your first name: Jesse
Please enter your last name: Huminski
```

Image 2.2: Prompt requesting information from the user.

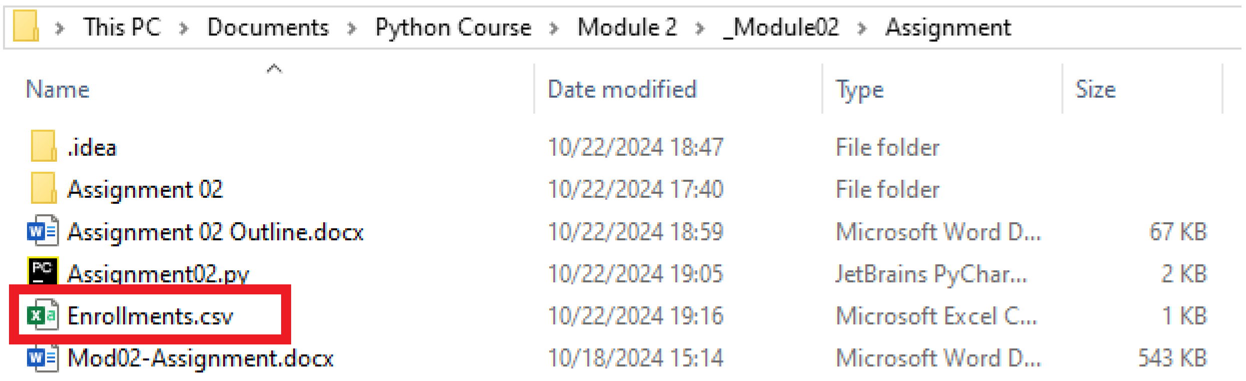
Upon entering my last name, the code prints the message I wrote and does the arithmetic properly. It then prints the variable "csv_data" to show that the data will accurately store in a CSV file. (Image 2.3)



```
Thank you, Jesse Huminski.
You have signed up for Python 100.
The cost of the course is 999.98.
Your total with tax is: 1089.98

Jesse,Huminski,Python 100,999.98,1089.98
```

Image 2.3: The program displaying accurate information to the user.

Finally, it is time to check that the data was written to the "Enrollments.csv" file correctly. After navigating to C:\Users\jesse\Documents\Python Course\Module 2_Module02\Assignment, I can see that a CSV file titled "Enrollments.csv" was generated by the program. (Image 2.4)

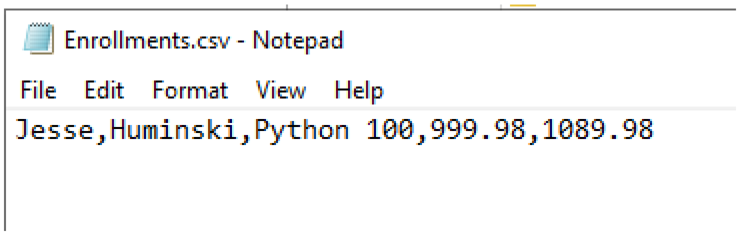


This PC > Documents > Python Course > Module 2 > _Module02 > Assignment

Name	Date modified	Type	Size
.idea	10/22/2024 18:47	File folder	
Assignment 02	10/22/2024 17:40	File folder	
Assignment 02 Outline.docx	10/22/2024 18:59	Microsoft Word D...	67 KB
Assignment02.py	10/22/2024 19:05	JetBrains PyChar...	2 KB
Enrollments.csv	10/22/2024 19:16	Microsoft Excel C...	1 KB
Mod02-Assignment.docx	10/18/2024 15:14	Microsoft Word D...	543 KB

Image 2.4: Enrollments.CSV generated by the program.

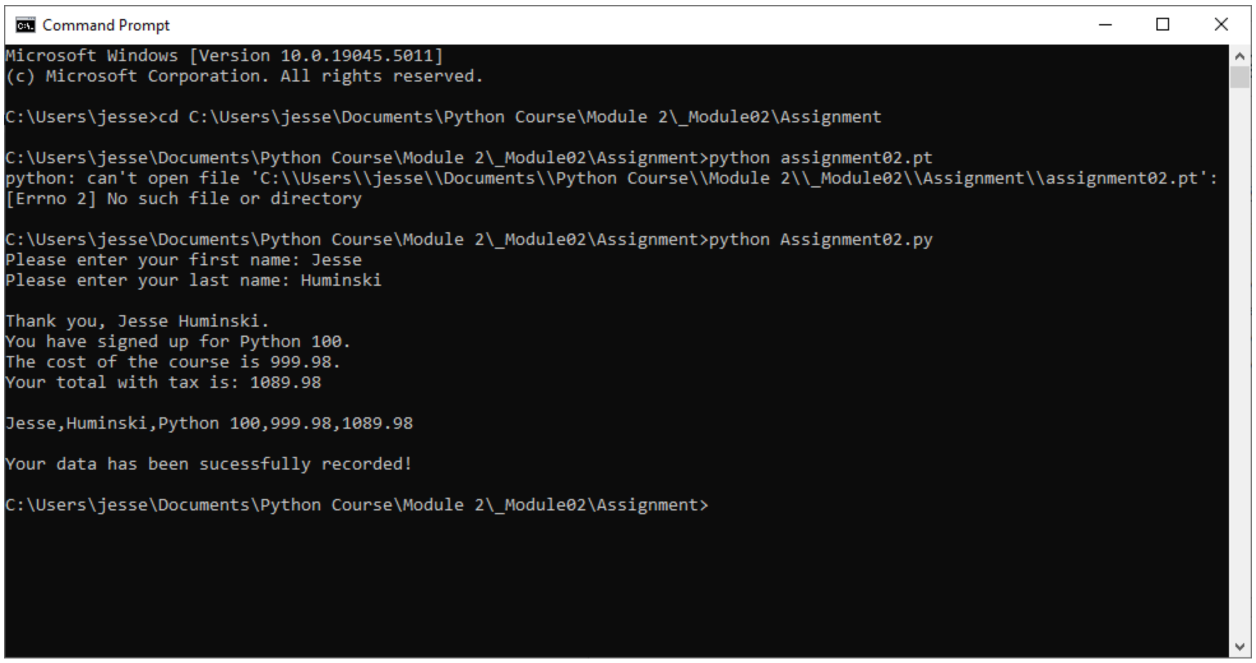
After opening the CSV file in notepad, I can see that the information is being written to the file accurately and without any bugs! (Image 2.5)



```
Jesse,Huminski,Python 100,999.98,1089.98
```

Image 2.5: Information from the program written to the CSV file.

Success!! One final step is to check that the program is working correctly in the Command Shell. (Image 2.6)



```
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jesse>cd C:\Users\jesse\Documents\Python Course\Module 2\_Module02\Assignment

C:\Users\jesse\Documents\Python Course\Module 2\_Module02\Assignment>python assignment02.pt
python: can't open file 'C:\\Users\\jesse\\Documents\\Python Course\\Module 2\\_Module02\\Assignment\\assignment02.pt':
[Errno 2] No such file or directory

C:\Users\jesse\Documents\Python Course\Module 2\_Module02\Assignment>python Assignment02.py
Please enter your first name: Jesse
Please enter your last name: Huminski

Thank you, Jesse Huminski.
You have signed up for Python 100.
The cost of the course is 999.98.
Your total with tax is: 1089.98

Jesse,Huminski,Python 100,999.98,1089.98

Your data has been sucessfully recorded!

C:\Users\jesse\Documents\Python Course\Module 2\_Module02\Assignment>
```

Image 2.6: Testing the program in the command shell

Summary

In Module 02, we dove deeper into the basics of the Python programming language. Moreso, in this module, we learned how to accurately gather data from the program's constant's and the user's inputs to write data to a file in CSV format. The information is received from the user and then printed onto the screen along with basic arithmetic functions. After the data is accrued, it is then stored into a CSV file to be called upon for future use.