**Module 3 Assignment**

Shashi Thakur

Colorado State University Global

Course Code: [23WC-CSC500-2](https://csuglobal.instructure.com/courses/88177)

Instructor - Dr Steven Evans

Due Date - 2/3/2024

Critical Thinking Assignment - Module 3 Part 1

1. Abstract of Assignment

**Part 1** - Write a program that calculates the total amount of a meal purchased at a restaurant. The program should ask the user to enter the charge for the food and then calculate the amounts with an 18 percent tip and 7 percent sales tax. Display each of these amounts and the total price.

1. Introduction

In module 3 we learned about Python data types and then zoomed into Arrays and iterated through arrays. In part of this assignment, the task is to develop a program which will read user input for the charge of food, do sales tax and tip calculation calculations and output total amount of tax, tips and total price. I plan to demonstrate how to read user inputs, do mathematical calculations and user proper formatting to display results for users. The part2 of the assignment will require a math library to do calculations.

1. Methodology

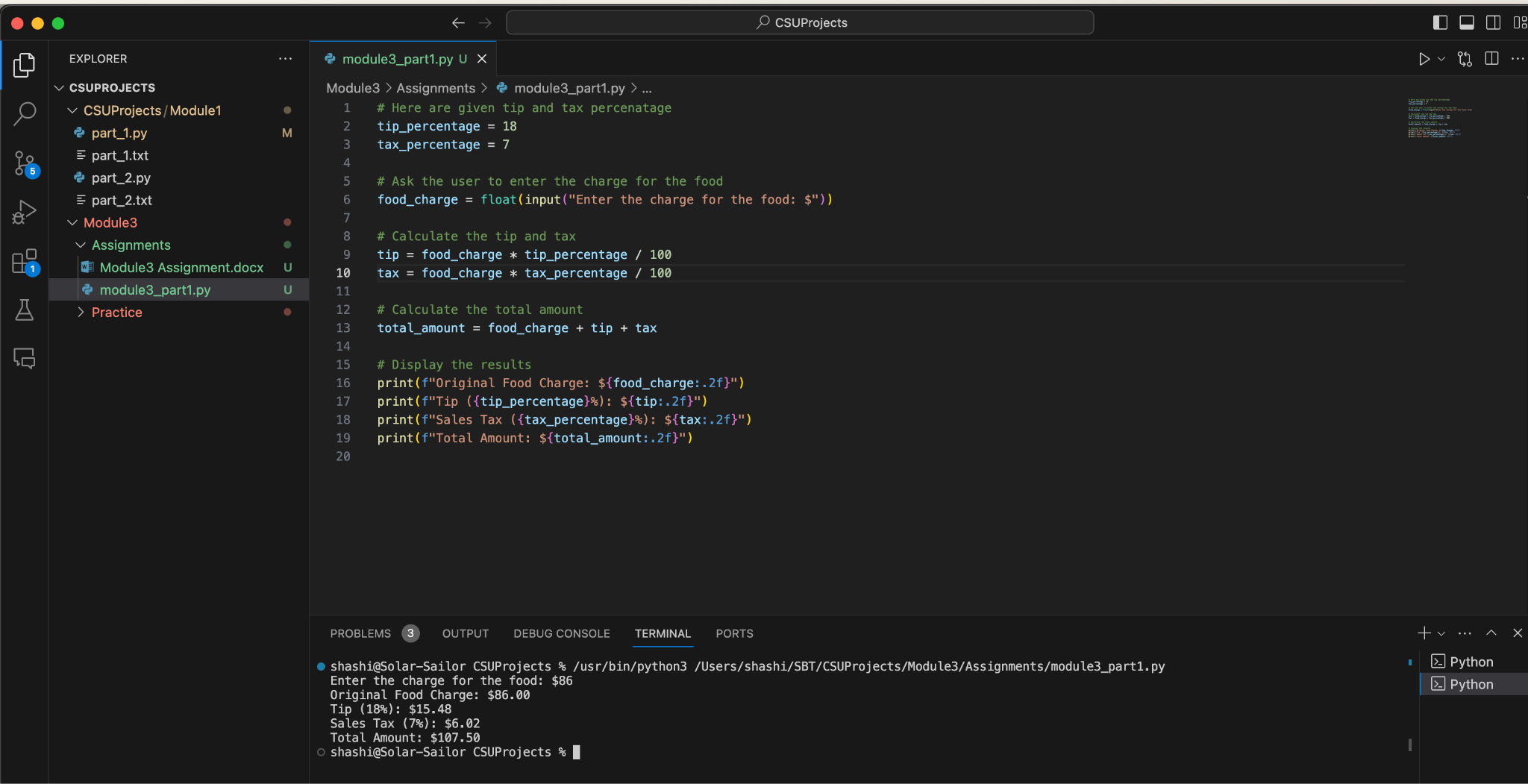
Ideally we should create a user interface for such programs. To keep matters simple, I will use it as a command prompt based utility. Here are the key considerations for this program.

1. The program should use float because we have to store price, sales tax etc
2. The calculations can be modularized so that it can easily handle changes in tax, tips etc
3. Considering future changes think about ability for user to enter tip percentage as input
4. Keep the program simple and minimalistic as per requirements
5. Implementation Details

Here the high level pseudocode based on the requirements

1. **Get user input: -**  Based on the requirements, there is only one input from the user. The program will ask user to enter charge for the food
2. **Calculate the tip and tax:-** Calculate the tip and tax based on the formula provided in requirement. Tip should be calculated at 18% and 7% for sales tax
3. **Calculate the total amount:-** Add the original charge, the tip, and the tax to get the total amount.
4. **Display the Results**: Show the user the original charge, the tip amount, the tax amount, and the total amount.

Screenshot of part1 code execution



1. Results and Discussion

The code addresses the requirements for part 1. I tested it using several different inputs. I have also parameterized the tip and tax percentage in the print statement so that any change in that is handled at one place only in the code.

Given an input of price for food, it calculates the tip and tax and prints these numbers for the user.It is able to handle float and integer.

After testing the code, I observed that it would be better if all the printed lines are right aligned so that all amounts show up aligned properly. It can be a challenge exercise. A right alignment logic can be implemented. I didn’t implement it at this point in time as it is out of scope.

If there is a need to get tip% from the user we can make minor changes in the code to work.

1. Supplemental Research

To make the output look closer to real receipt I did supplemental research on alignment.

This document provides a good explanation on alignment - <https://www.geeksforgeeks.org/string-alignment-in-python-f-string/>

The approach suggested is using the f-strings to format the text. The syntax of the alignment of the output string is defined by ‘<‘, ‘>’, ‘^’ and followed by the width number.

1. Conclusion

Data structures are fundamental to any programming language and any programming we do. Python . It allows efficient organization of information, its retrieval and any manipulations. Python has a variety of built in data structures. We learned about List, Tuple, Set, Dictionary, String. We also learned basic usage of the math library and how formatting can be done for desired outputs.

1. References

* <https://www.geeksforgeeks.org/string-alignment-in-python-f-string/>
* <https://docs.python.org/3/tutorial/datastructures.html>
* <https://www.geeksforgeeks.org/python-data-structures/>
* <https://www.geeksforgeeks.org/calculate-time-difference-in-python/>

1. Appendices

* Full code listings - .py files attached as separate files in assignment
* Github - <https://github.com/shashithakurcsu/CSUProjects/blob/main/Module3/Assignments/module3_part1.py>

### **Challenges & Experiences in this Module:**

* Challenge: Formatting of outputs.
  + Experience: I researched and practiced several ways of formatting output for users. The ZyBooks exercises were very helpful but reading and seeing examples from supplemental research was very helpful

Critical Thinking Assignment - Module 3 Part 2

1. Abstract of Assignment

**Part 2** - Many people keep time using a 24-hour clock (11 is 11am and 23 is 11pm, 0 is midnight). If it is currently 13 and you set your alarm to go off in 50 hours, it will be 15 (3pm). Write a Python program to solve the general version of the above problem. Ask the user for the time now (in hours) and then ask for the number of hours to wait for the alarm. Your program should output what the time will be on a 24-hour clock when the alarm goes off.

1. Introduction

In module 3 we learned about Python data types and then zoomed into Arrays and iterated through arrays. In part 2 of this assignment, the task is to develop a program which will read user inputs.

1. Current time in hours
2. Numbers of hours to wait for alarm

Given these input the program will calculate and display the the time when alarm will go off on a 24-hour clock

The program should also outline clearly that input is requested hour in 24 hour clock format and output is displayed in the same format.

1. Methodology

To solve the problem in this task, we have to develop a Python program which will do the following

1. Obtain input from user for the current time in hours on a 24-hour clock format
2. Obtain input from user for no of hours to wait for the alarm
3. Calculate the time when alarm will go off in 24 hour clock format

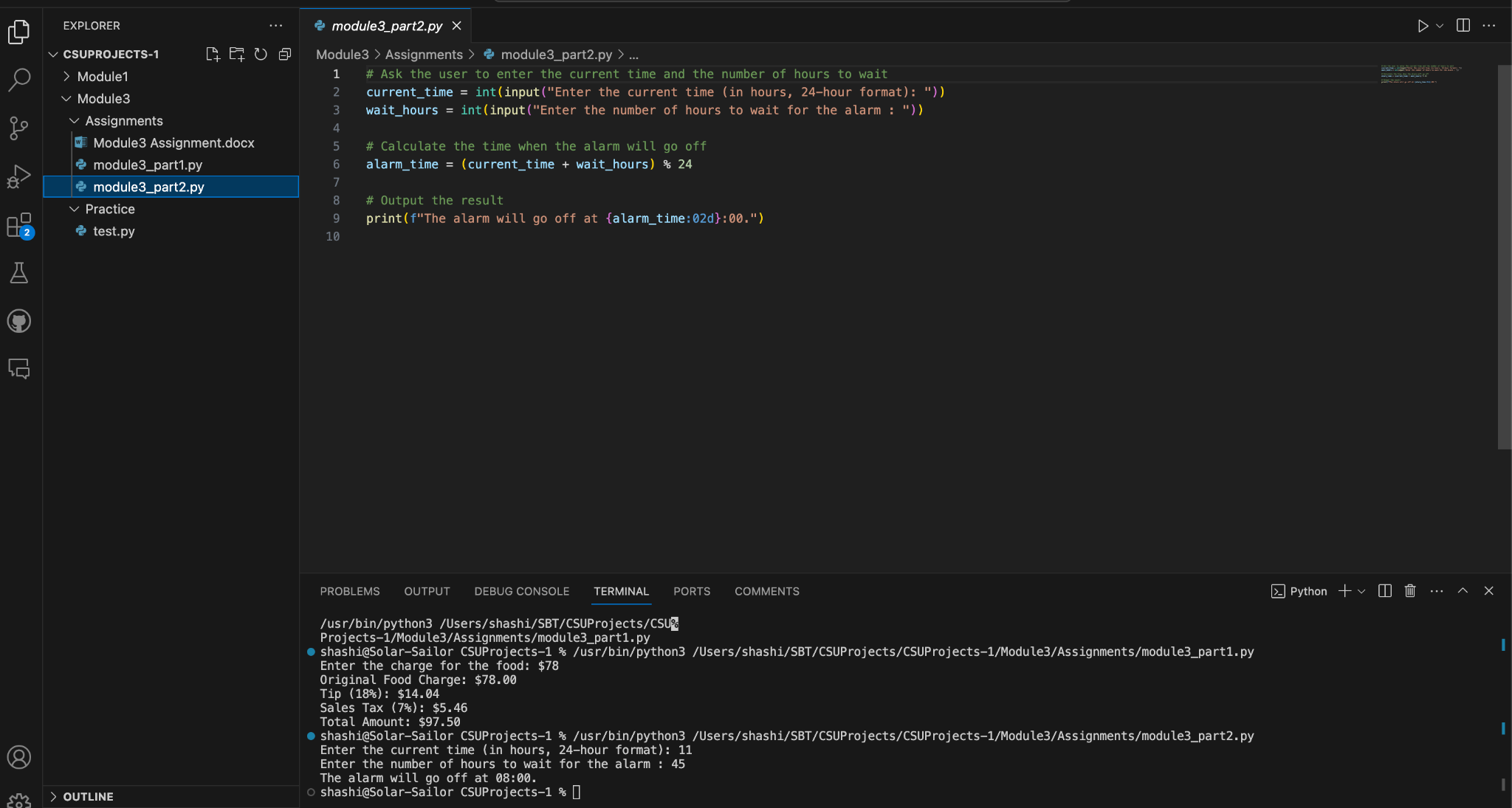
Taking an example given in the problem statement, let's assume that the current time is 1PM or 13 and the wait time is 50 hours. So, the alarm clock should go off at 13 + 24+ 24 + 2 hours which is at 15 hours. We have to implement this logic in Python. This can be done by dividing the wait time by 24 and check remainder. The remainder should be added to current time. To avoid another division by 24 it's simpler and efficient to add current time and wait time and then apply modulo operator.

1. Implementation Details

Here the high level pseudocode based on the requirements

1. **Get current time as input: -** 
   1. The user is asked to enter the current time in hour. Prompt user to enter current time in 24-hour format.
   2. Store this input in a variable current\_time
2. **Get wait hours as input:-**
   1. The user is asked how many hours they want to wait for the alarm to go off.
   2. This has to be a non-negative integer. It should be stated in the input statement
   3. The input is stored in a variable wait\_hours
3. **Calculate alarm time**
   1. The program adds current time and wait time. Since a day has 24 hours, the program applies modulo 24 on the sum of current time and wait time.
4. **Display Results**
   1. The program displays the time when the alarm will go off.

Screenshot of part1 code execution



1. Results and Discussion

The code addresses the requirements for part 2. I tested it using several different inputs.

After testing the code, This program can be improved by making the program handle hours and minutes as well. Say for example, user may enter 13:45

In this case we will have to split input time in hours and minutes. Calculate %24 and then add remaining minutes to the result. If we allow user to enter alarm time in hours and minutes then we will have to create a function to handle addition of hours and minutes.

1. Supplemental Research

I found that we can use the Python datetime library to handle hours, minutes and seconds calculations easily.

1. Conclusion

Data structures are fundamental to any programming language and any programming we do. Python . It allows efficient organization of information, its retrieval and any manipulations. Python has a variety of built in data structures. We learned about List, Tuple, Set, Dictionary, String. We also learned basic usage of the math library and how formatting can be done for desired outputs.

1. References

* <https://www.geeksforgeeks.org/string-alignment-in-python-f-string/>
* <https://docs.python.org/3/tutorial/datastructures.html>
* <https://www.geeksforgeeks.org/python-data-structures/>

1. Appendices

* Full code listings - .py files attached as separate files in assignment
* Github link - <https://github.com/shashithakurcsu/CSUProjects/blob/main/Module3/Assignments/module3_part2.py>

### **Challenges & Experiences in this Module:**

* Challenge: Formatting of outputs.
  + Experience: I researched and practiced several ways of formatting output for users. The ZyBooks exercises were very helpful but reading and seeing examples from supplemental research was very helpful