##

Student: Aditya Singh Sandhu

Course: CSC 500 – Principles of Programming

Module: 3 - Critical Thinking Assignment

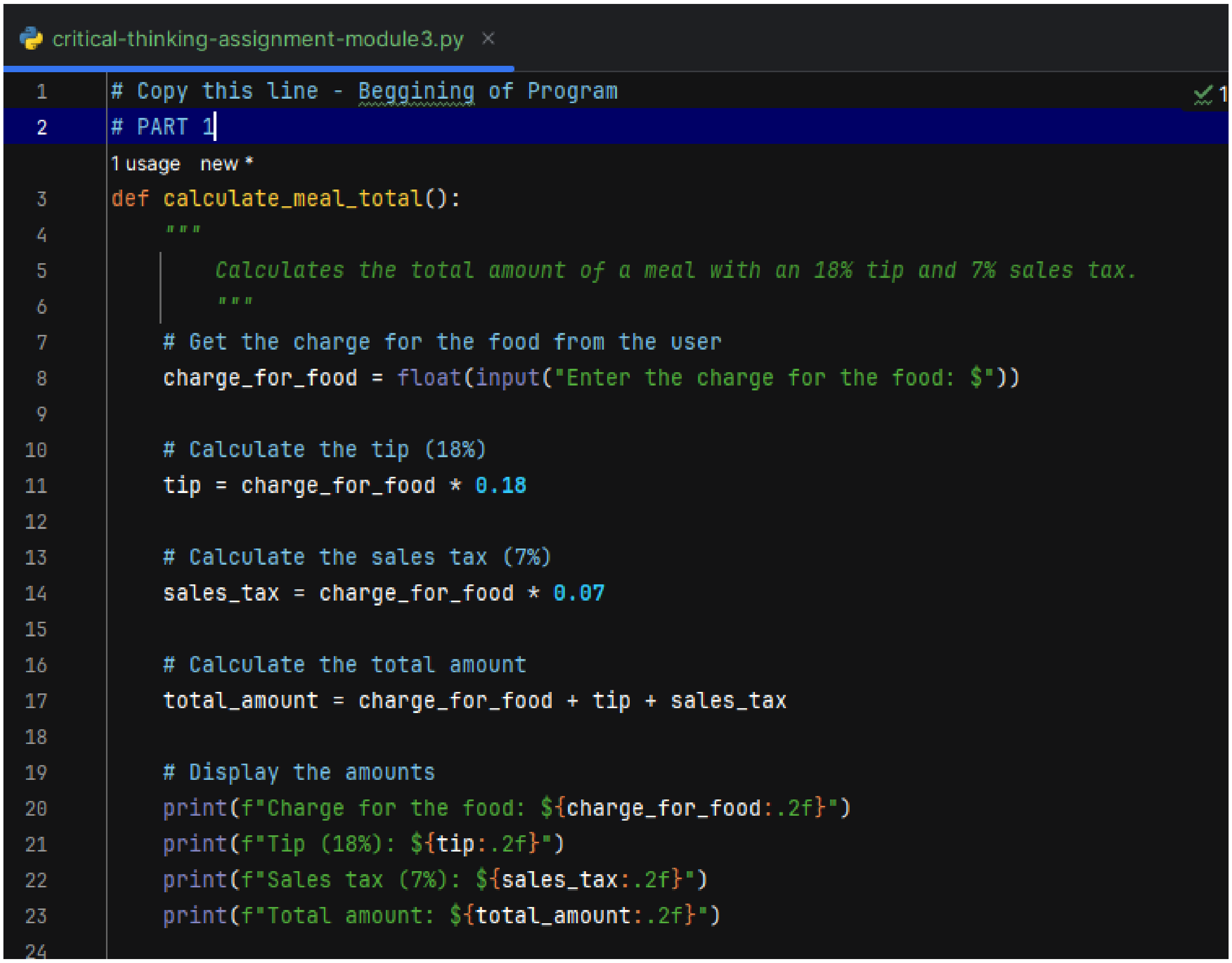
##

Python3 Code ~

```
# Copy this line - Beginning of Program

# PART 1

def calculate_meal_total():
    """
        Calculates the total amount of a meal with an 18% tip and 7% sales tax.
        """
    charge_for_food = float(input("Enter the charge for the food: $"))
    tip = charge_for_food * 0.18
    sales_tax = charge_for_food * 0.07
    total_amount = charge_for_food + tip + sales_tax

    print(f"Charge for the food: ${charge_for_food:.2f}")
    print(f"Tip (18%): ${tip:.2f}")
    print(f"Sales tax (7%): ${sales_tax:.2f}")
    print(f"Total amount: ${total_amount:.2f}")



# PART 2

def calculate_alarm_time():
    """
    Calculates the time when an alarm will go off given the current time and a wait time in hours.
    """
    current_time = int(input("Enter the current time (in 24-hour format): "))
    hours_to_wait = int(input("Enter the number of hours to wait for the alarm: "))
    alarm_time = (current_time + hours_to_wait) % 24
    print(f"The alarm will go off at: {alarm_time:02d}:00")


def main():
    """
    Main function to run the meal total calculation and alarm time calculation.
    """
    # PART 1
    print("Meal Total Calculator")
    calculate_meal_total()
    # PART 2
    print("\nAlarm Time Calculator")
    calculate_alarm_time()


# Call the main function
# RUN ENTIRE PROGRAM
if __name__ == "__main__":
    main()
```

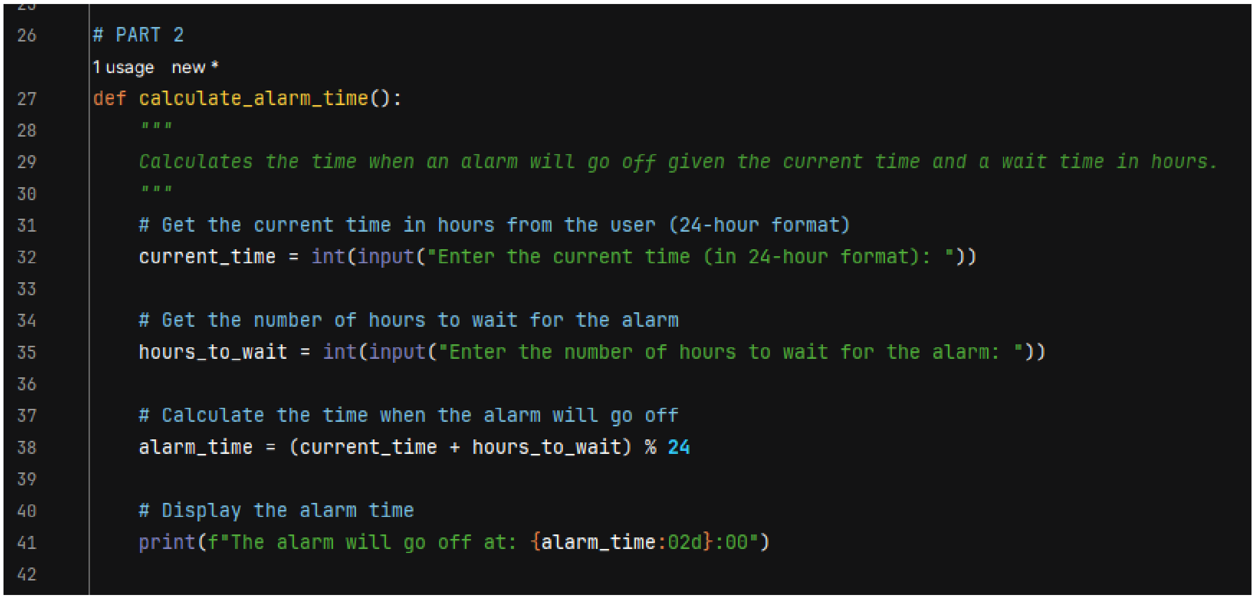
```
# Copy this line - End of Program
```

"Screenshot 1," a Python function named `calculate_meal_total` is defined to compute the total cost of a meal, including an 18% tip and a 7% sales tax. The function prompts the user to input the charge for the food and calculates the tip and sales tax based on the given percentages. The total amount, inclusive of the meal charge, tip, and tax, is then calculated and displayed. The function employs `float` to convert the input into a floating-point number and uses formatted string literals (f-strings) to print the calculated values, rounded to two decimal places.



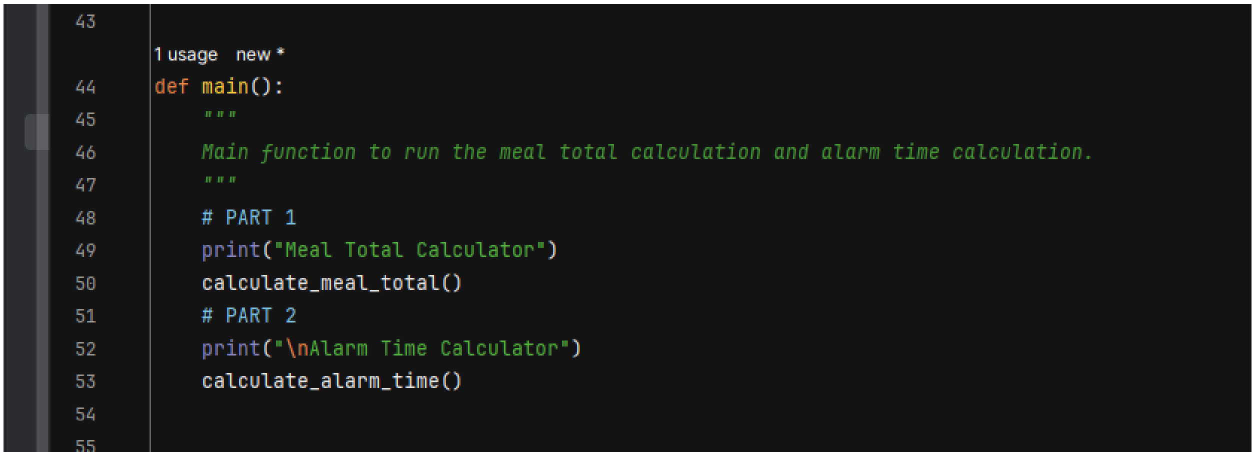
```python
# Copy this line - Beggining of Program
# PART 1
def calculate_meal_total():
    """
        Calculates the total amount of a meal with an 18% tip and 7% sales tax.
        """
    # Get the charge for the food from the user
    charge_for_food = float(input("Enter the charge for the food: $"))

    # Calculate the tip (18%)
    tip = charge_for_food * 0.18

    # Calculate the sales tax (7%)
    sales_tax = charge_for_food * 0.07

    # Calculate the total amount
    total_amount = charge_for_food + tip + sales_tax

    # Display the amounts
    print(f"Charge for the food: ${charge_for_food:.2f}")
    print(f"Tip (18%): ${tip:.2f}")
    print(f"Sales tax (7%): ${sales_tax:.2f}")
    print(f"Total amount: ${total_amount:.2f}")
```

"Screenshot 2," a Python function named `calculate_alarm_time` is presented. This function calculates the time at which an alarm will go off based on the current time and the number of hours to wait, as provided by the user. The function prompts the user to enter the current time in a 24-hour format and the number of hours to wait for the alarm. It then calculates the alarm time using the modulo operation to ensure the time is within the 24-hour format range. The result is displayed in a formatted string, indicating the time the alarm will ring.



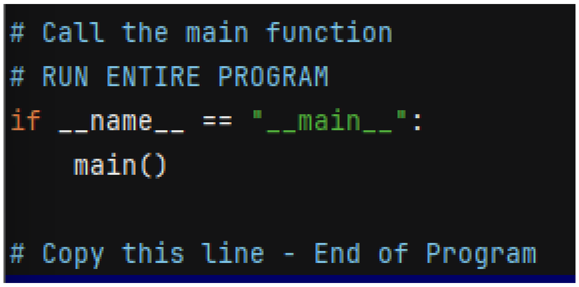
```python
# PART 2
def calculate_alarm_time():
    """
    Calculates the time when an alarm will go off given the current time and a wait time in hours.
    """
    # Get the current time in hours from the user (24-hour format)
    current_time = int(input("Enter the current time (in 24-hour format): "))

    # Get the number of hours to wait for the alarm
    hours_to_wait = int(input("Enter the number of hours to wait for the alarm: "))

    # Calculate the time when the alarm will go off
    alarm_time = (current_time + hours_to_wait) % 24

    # Display the alarm time
    print(f"The alarm will go off at: {alarm_time:02d}:00")
```

"Screenshot 3," a Python function named `main()` is defined. This function serves as the entry point for the program, orchestrating the execution of two previously defined functions: `calculate_meal_total()` and `calculate_alarm_time()`. The `main()` function prints introductory text for each part—"Meal Total Calculator" and "Alarm Time Calculator"—and then calls the respective functions. The purpose of `main()` is to provide a structured flow to the program, guiding the user through the process of calculating a meal's total cost and determining the time an alarm will go off.



```
def main():
    """
    Main function to run the meal total calculation and alarm time calculation.
    """
    # PART 1
    print("Meal Total Calculator")
    calculate_meal_total()
    # PART 2
    print("\nAlarm Time Calculator")
    calculate_alarm_time()
```

"Screenshot 4," the final segment of the Python program is displayed. This segment includes a conditional statement that checks if the script is being run as the main module. The line ` if __name__ == "__main__":` ensures that the ` main()` function is called only when the script is executed directly, rather than being imported as a module in another script. This is a common Python idiom that allows the program to define functions and classes and run the script as a standalone program. Below this conditional, the ` main()` function is called, thereby executing the entire program.



```
# Call the main function
# RUN ENTIRE PROGRAM
if __name__ == "__main__":
    main()

# Copy this line - End of Program
```

"Screenshot 5," the output of the program's ` calculate_meal_total` function is displayed in the console. The user is prompted to "Enter the charge for the food," with an example value of ` $150.000` entered. The program calculates and displays the following:

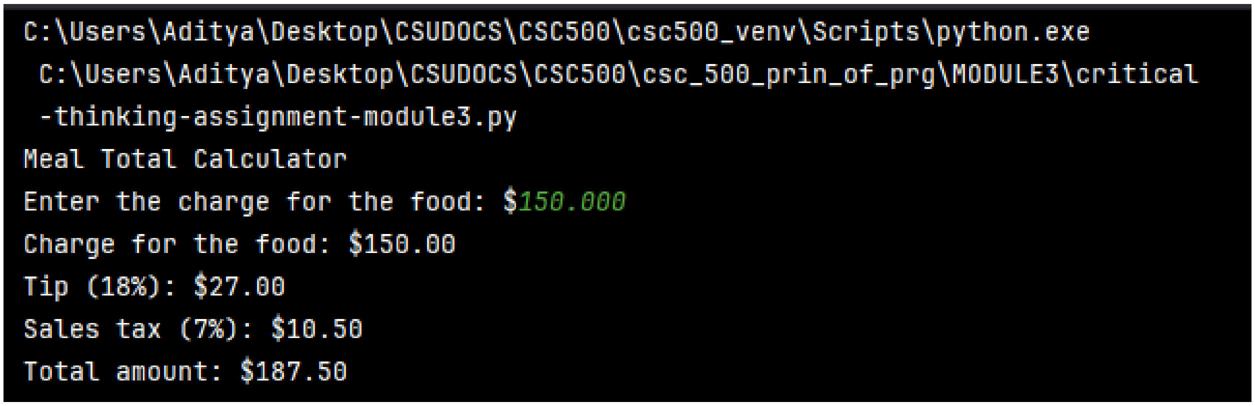
- Charge for the food: $150.00

- Tip (18%): $27.00

- Sales tax (7%): $10.50

- Total amount: $187.50

This output shows the breakdown of the meal cost, including the original charge, calculated tip, sales tax, and the total amount due. The program correctly rounds and formats the monetary values to two decimal places for clarity.



```
C:\Users\Aditya\Desktop\CSUDOCS\CSC500\csc500_venv\Scripts\python.exe
 C:\Users\Aditya\Desktop\CSUDOCS\CSC500\csc_500_prin_of_prg\MODULE3\critical
 -thinking-assignment-module3.py
Meal Total Calculator
Enter the charge for the food: $150.000
Charge for the food: $150.00
Tip (18%): $27.00
Sales tax (7%): $10.50
Total amount: $187.50
```

"Screenshot 6," the output of the program's `calculate_alarm_time` function is displayed. The program prompts the user to:

1. Enter the current time (in 24-hour format): The user inputs 24.

2. Enter the number of hours to wait for the alarm: The user inputs 12.

The program then calculates the alarm time and displays:

- The alarm will go off at: 12:00

This indicates that if the current time is midnight (represented as 24 or 0 in 24-hour format), and the alarm is set for 12 hours later, it will go off at 12:00 PM. The program successfully computes the time in a 24-hour format and indicates completion with "Process finished with exit code 0," signifying that the program terminated successfully without errors.

References

GeeksforGeeks. (n.d.). *Python data types*. Retrieved from https://www.geeksforgeeks.org/python-data-types/

GeeksforGeeks. (n.d.). *Python arrays*. Retrieved from https://www.geeksforgeeks.org/python-arrays/

TutorialsPoint. (n.d.). *Numpy - Iterating over array*. Retrieved from https://www.tutorialspoint.com/numpy/numpy_iterating_over_array

PythonBasics. (n.d.). *Variables in Python*. Retrieved from https://pythonbasics.org/variables/

TutorialsPoint. (n.d.). *What is a sequence data type in Python?* Retrieved from https://www.tutorialspoint.com/What-is-a-sequence-data-type-in-Python

TutorialsPoint. (n.d.). *Computer programming - Arrays*. Retrieved from https://www.tutorialspoint.com/computer_programming/computer_programming_arrays.htm