

#####

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():

 food_cost = float(input("food charge: \$"))

 tip = food_cost * 0.18

 sales_tax = food_cost * 0.07

 total_amount = food_cost + tip + sales_tax

 print(f"Cost of the food: \${food_cost:.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():

 current_time = int(input("Enter current time (in 24-hour format): "))

 hours_to_wait = int(input("Enter count of hours for the alarm: "))

```
alarm_time = (current_time + hours_to_wait) % 24  
print(f"The alarm goes 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
```

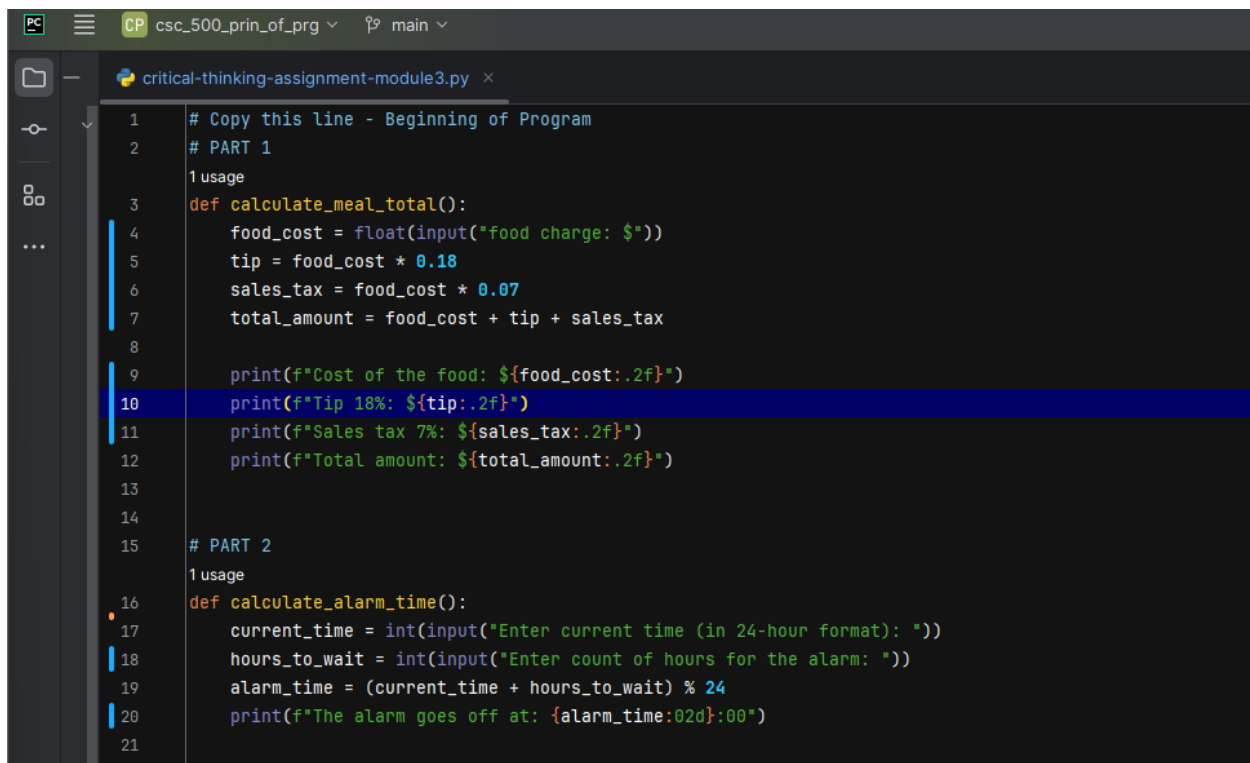
```
# RUN ENTIRE PROGRAM
```

```
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. The function prompts the user to input the cost for the food and makes the relevant calculations 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.

A screenshot of a Python IDE window titled 'critical-thinking-assignment-module3.py'. The code is divided into two sections: PART 1 and PART 2. PART 1 contains a function 'calculate_meal_total()' which prompts for food cost, calculates tip (18%) and sales tax (7%), and prints the total amount. PART 2 contains a function 'calculate_alarm_time()' which prompts for current time and hours to wait, then calculates the alarm time using modulo 24. Line 10 is highlighted in blue.

```
1 # Copy this line - Beginning of Program
2 # PART 1
3 1 usage
4 def calculate_meal_total():
5     food_cost = float(input("food charge: $"))
6     tip = food_cost * 0.18
7     sales_tax = food_cost * 0.07
8     total_amount = food_cost + tip + sales_tax
9
10    print(f"Cost of the food: ${food_cost:.2f}")
11    print(f"Tip 18%: ${tip:.2f}")
12    print(f"Sales tax 7%: ${sales_tax:.2f}")
13    print(f"Total amount: ${total_amount:.2f}")
14
15 # PART 2
16 1 usage
17 def calculate_alarm_time():
18     current_time = int(input("Enter current time (in 24-hour format): "))
19     hours_to_wait = int(input("Enter count of hours for the alarm: "))
20     alarm_time = (current_time + hours_to_wait) % 24
21     print(f"The alarm goes off at: {alarm_time:02d}:00")
```

"Screenshot 2," a Python function named 'calculate_alarm_time' is presented. 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 range.

The result is displayed in a formatted string, indicating the time the alarm will ring.

```

# PART 2
1 usage
def calculate_alarm_time():
    current_time = int(input("Enter current time (in 24-hour format): "))
    wait_time = int(input("Enter count of hours for the alarm: "))
    alarm_time = (current_time + wait_time) % 24
    print(f"The alarm is set to 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.

```

1 usage
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.

```

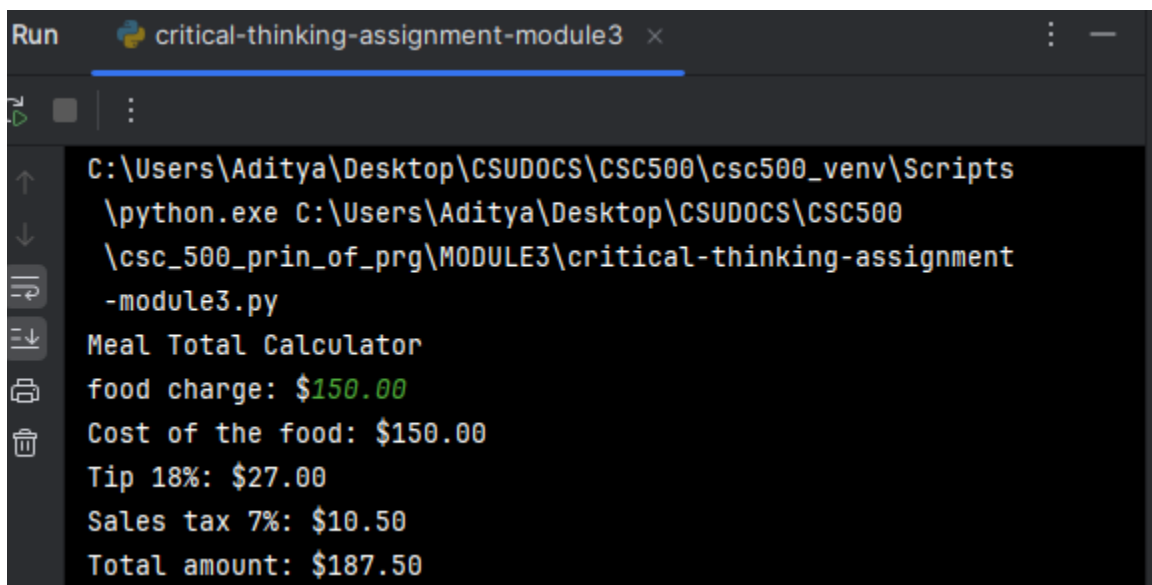
4
5 # Call the main function
6 # RUN ENTIRE PROGRAM
7 ► if __name__ == "__main__":
8     main()
9
10 # Copy this line - End of Program
11

```

"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.

A screenshot of a terminal window titled "Run" with a tab labeled "critical-thinking-assignment-module3". The terminal shows the command to run a Python script: `C:\Users\Aditya\Desktop\CSUD0CS\CSC500\csc500_venv\Scripts\python.exe C:\Users\Aditya\Desktop\CSUD0CS\CSC500\csc_500_prin_of_prg\MODULE3\critical-thinking-assignment-module3.py`. The output of the program is displayed below the command: `Meal Total Calculator`, `food charge: $150.00`, `Cost of the food: $150.00`, `Tip 18%: $27.00`, `Sales tax 7%: $10.50`, and `Total amount: $187.50`. The terminal has a dark background with light-colored text.

"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. number of hours to wait for the alarm to go off: The user inputs 12.
- 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.

```
Alarm Time Calculator
Enter current time (in 24-hour format): 24
Enter count of hours for the alarm: 12
The alarm is set to go off at: 12:00

Process finished with exit code 0
```

GITHUBLINK -

[csc_500_prin_of_prg/MODULE3/critical-thinking-assignment-module3-aditya-sandhu.pdf](https://github.com/65AR645ASAN/csc_500_prin_of_prg/blob/main/MODULE3/critical-thinking-assignment-module3-aditya-sandhu.pdf)
at main · 65AR645ASAN/csc_500_prin_of_prg (github.com)

References

Python Land. (n.d.). Python float. Retrieved August 3, 2024, from https://python.land/python-data-types/python-float#google_vignette

Python Basics. (n.d.). Strings in Python. Retrieved August 3, 2024, from <https://pythonbasics.org/strings/>

TutorialsPoint. (n.d.). Dictionary data type in Python. Retrieved August 3, 2024, from <https://www.tutorialspoint.com/dictionary-data-type-in-python>