Module 3 Critical Thinking Assignment

Figure 1

Pseudocode for Restaurant Bill and Alarm Clock

```
START
FUNCTION get_input_for_meal() RETURNS FLOAT
    TRY
        PROMPT and STORE for meal charge input (force float)
        RETURN meal_charge
    EXCEPT ValueError
        PRINT “Invalid. Must be an integer.”
        CALL get_input_for_meal() again
FUNCTION calculate_meal(meal_charge: FLOAT) RETURNS None
    Tax = meal_charge * 0.07
    Tip = meal_charge * 0.18
    Total = meal_charge + tax + tip
    PRINT “7% sales tax: $” format(tax, 2)
    PRINT “18% tip: $” format(tip, 2)
    PRINT “Total amount: $” format(total, 2)
FUNCTION get_input_for_clock() RETURNS tuple[int, int]
    TRY
        PROMPT for current time in hours and STORE in user_input_time_hours
        PROMPT user for time to wait and STORE in user_input_wait_time
        RETURN user_input_time_hours, user_input_wait_time
```

```
        EXCEPT ValueError
            Print “Invalid. Must be an integer”
            CALL get_input_for_clock()
FUNCTION clock(user_input_time_hours: int, user-input_wait_time: int) RETURNS None
        Seat new_time = ((user_input_wait_time % 24) + user_input_time_hours) % 24
        PRINT “In” + user_input_wait_time + “ hours it will be “ .format(new_time, 2) + “:00”
FUNCTION main() returns None
        Set user_input_charge = CALL get_input_for_meal()
        CALL calculate_meal(user_input_charge)
        SET user_input_time_hours, user_input_wait_time = CALL get_input_for_clock
        CALL clock(user_input_time_hours, user_input_wait_time)
END
```

Note. This pseudocode illustrates the algorithm used in the Python script for calculating a simple restaurant bill and alarm clock. The bill calculates the tax at 7%, tip at 18% and then the total. The alarm clock receives the current time in hours from the user, how long to wait or set the alarm for, and outputs the hour on a 24-hour clock when the alarm will go off.

Figure 2

Source Code for Restaurant Bill and Alarm Clock

```
# File Name: Cline_Jason_RestaurantBill_AlarmClockW3.py

# Author: Jason Todd Cline

# Institution: Colorado State University Global
# Class: CSC500-1
# Term: 24FB
# Module: 3

# Date Created: 09/08/2024
# Last Modified: 09/08/2024


def get_user_input_meal() -> float:
    """
    Get user input for the charge of the food.

    Returns:
        float: The charge for the food.
    """
    try:
        meal_charge = float(input("Enter the charge for the food: "))
        return meal_charge
    except ValueError:
        print("Invalid input. Please enter a valid number.")
        return get_user_input_meal()


def calculate_meal(meal_charge: float) -> None:
    """
    Calculate the tax, tip, and total amount for the meal.

    Args:
        meal_charge (float): The charge for the food.
    """
    tax = meal_charge * 0.07
```

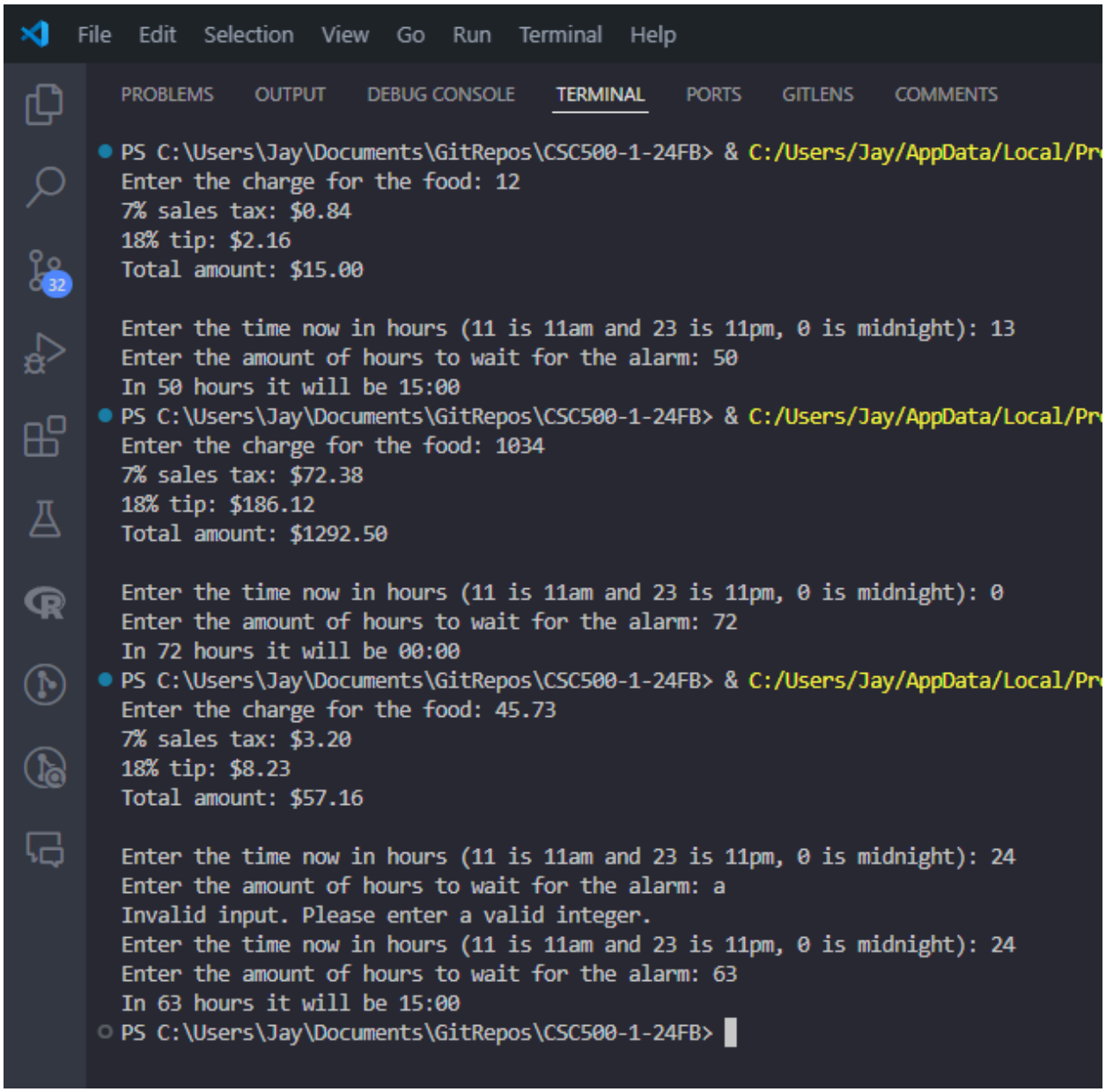
```
    tip = meal_charge * 0.18
    total = meal_charge + tax + tip
    print(f"7% sales tax: ${tax:.2f}")
    print(f"18% tip: ${tip:.2f}")
    print(f"Total amount: ${total:.2f}\n")


def get_user_input_clock() -> tuple[int, int]:
    """
    Get user input for the current time and wait time for the alarm.

    Returns:
        tuple[int, int]: The current time and wait time for the alarm.
    """
    try:
        user_input_time = int(
            input(
                "Enter the time now in hours (11 is 11am and 23 is 11pm, 0
is midnight): "
            )
        )
        user_input_wait_time = int(
            input("Enter the amount of hours to wait for the alarm: ")
        )
        return user_input_time, user_input_wait_time
    except ValueError:
        print("Invalid input. Please enter a valid integer.")
        return get_user_input_clock()


def clock(user_input_time: int, user_input_wait_time: int) -> None:
    """
    Calculate the new time after waiting for the specified hours.

    Args:
        user_input_time (int): The current time.
        user_input_wait_time (int): The amount of hours to wait for the
alarm.
    """
    new_time = ((user_input_wait_time % 24) + user_input_time) % 24
```

```
    print(f"In {user_input_wait_time} hours it will be {new_time:02d}:00")


def main() -> None:
    """
    Main function to run the program.
    """
    user_input_charge = get_user_input_meal()
    calculate_meal(user_input_charge)
    user_input_time, user_input_wait_time = get_user_input_clock()
    clock(user_input_time, user_input_wait_time)


if __name__ == "__main__":
    main()
```

Note. This figure displays the source code used for a Python script that calculates a restaurant bill and provides the user an alarm clock based on input. The script prompts for user input to calculate the tax, tip and total amount for a meal. It also accepts user input for the current time and wait time for an alarm, displaying the new time after some specified interval.

Figure 3

Execution and Testing for Restaurant Bill and Alarm Clock



Note. Python output of a restaurant bill and alarm clock that demonstrates fetching input, basic calculations (addition, multiplication, modulus), error-handling, and string formatting.

References

Cline, J. T. [Jay4rmTheBay]. (2024). *CSC500-1-24FB* [Source code]. GitHub. https://github.com/Jay4rmTheBay/CSC500-1-24FB