Name:

Luke Seavey

Date Created:

08/23/25

Program Description:

This program is a simple ticket sales counter that tracks the amount of buyers for a set amount of tickets (40) and ends sales when all are sold with a max of 4 tickets per person.

Functions used in the Program (list in order as they are called):

1. Function Name: selling_tickets

Description:

Main handler for ticket sales, handles sales, buyer count, and error handling i.e buying more then 4 tickets

Parameters:

- num_tix, this tracks how many tickets the buyer wants to buy.

Variables:

- Total_tix (int, global) – remaining tickets.
- max_per_person (int, global) – purchase limit per buyer (4).
- sales (int, global) – count of successful buyers.
- num_tix (int, parameter) – the number of requested tickets.

Logical Steps:

1. If num_tix > max_per_person: reject with a message about the 4-ticket limit.
2. Else if num_tix > total_tix: reject with a “not enough tickets” message.
3. Else: subtract num_tix from total_tix, increment sales by 1, and return a success message including num_tix.

Returns:

- Returns a string regarding if the amount of tickets are available or if the buyer tries to buy more than 4. If buyer tickets are valid then returns a string with the amount of ticks bought

2. Function Name: main

Description:

Main while loop to keep selling tickets until they are sold out.

Parameters:

- There are no parameters in main

Variables:

- Num_tix (int, local) - It is an integer that is used for how many tickets are available and how many would you like to buy.
- Message (str, local) - is a string that calls the function of the number of tickets and prints how many they buy.

Logical Steps:

1. While total_tix > 0:
    a. try to read an integer from the user for num_tix.
    b. Call selling_tickets(num_tix) and store its return in a message.
    c. Print message and the updated “tickets remaining.”
    d. On ValueError, print “Please enter a valid number.” and continue.
2. After the loop ends (sold out), print “All tickets sold out! Total buyers: {sales}”.

Returns:

No, value is returned as outputs to the terminal

Logical Steps:

1. The interpreter executes the guard if name == "main": and calls main().
2. Inside main(), for each purchase attempt, the program calls selling_tickets(num_tix) to process the request and update state.
3. The loop ends when total_tix reaches 0, then the program prints the final summary.

List the order in which your functions are called.

1. Program start main()
2. Inside main(), for each loop iteration:
    - Read num_tix
    - Call selling)tickets(num_tix)
    - Print the returned message
3. Repeat step 2 until tickets reach 0, then exit.

Link to your repository:

https://github.com/FennecAce/COP2373/tree/94542fa92efd6cfb69086c3ddfb1df60ebc03b41/LukeSeavey_ProgramingExercse_1

Output Screenshot: