**Name**: Jason Miller

**Date Created**: January 13, 2025

**Program Description:**

This program attempts to sell 20 cinema tickets and allows users to pre-order up to four tickets. Lastly, it will display the total number of buyers.

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

1. **Function Name**: per\_customer(ticket)

**Description**: This function loops if a user attempts to purchase more than four tickets at a time. Once a user buys a valid number of tickets, it returns that value.

**Parameters**: purchase (input) – A valid number of tickets

**Variables**: purchase (input) – Used as a conditional statement to loop function and receives a number of tickets a user wishes to purchase

**Logical** Steps:

1. Detects if purchase is greater than 4 (initially it already is).
2. Loops if “purchase” is greater than 4.
3. Tell users they are limited to 4 tickets per customer.
4. Prompts the user to enter a valid number of tickets
5. If the user still does not enter a valid number of tickets, the loop will detect this and reactivate
6. Return valid input

**Returns**: purchase (input): variable with a valid number of tickets.

2. **Function Name**: get\_tickets ()

**Description**: This function sells a valid number of tickets and counts a valid number of customers until all tickets are sold.

**Parameters**: None

**Variables**:

1. ticket (decumulator?/condition statement) – Used to determine if any more tickets need to be bought. Also displays to the user the number of tickets left.
2. buy (input) – Used to get the number of tickets a user wants to purchase
3. limit (integer) – Result of the “per\_customer” function
4. buyer (accumulator) – Holds the amount of valid customers

**Logical** **Steps**:

1. Loops if there are still tickets to be sold.
2. Prompts the user to purchase tickets.
3. Cancels the purchase if a customer attempts to purchase zero tickets
4. Calls the per\_customer function if the customer attempts to purchase more than 4 tickets
   1. Sells valid number of tickets
   2. Cancels the purchase if a customer attempts to purchase more than the existing number of tickets
   3. Adds 1 to the buyer variable
   4. Displays the tickets

OR

1. Sells valid number of tickets
2. Adds 1 to the buyer variable
3. Cancels the purchase if a customer attempts to purchase more than the existing number of tickets

Displays the tickets

1. Displays the amount of valid customers

**Returns**: None

**Logical Steps:**

1. Call the get\_tickets function
2. The per\_customer is called within the get\_tickets funtion

[My repository](https://github.com/JMiller220/COP2373/tree/master/Assignments)