**Technical Design Document**

**Name:** Zachary Phillips

**Date Created:** August 31, 2025

**Program Description:**

Pre-sells a set amount of cinema tickets. Each buyer can purchase up to 4 tickets for transaction. The program loops and prompts for input on how many tickets to purchase per that instance of a buyer, validates the request in the loop, updates the standing value for tickets and buyers, and then updates the number of tickets left after each purchase. Once the tickets have sold, output the number of buyers that purchased the tickets.

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

1. **Function Name:** main

**Description:**

Program flow with starting the fixed inventory at 20 tickets, then calls the sale logic, and prints the final output

**Parameters:** (none)

**Variables:**

buyer\_counter (int) – holds current count for amount of buyers that comes from ticket\_logic

**Logical Steps:**

1. Initialize the sale, calls ticket\_logic(20) for purchases
2. Storage of the buyer count in buyer\_counter
3. Call display\_now(buyer\_counter) to print summary at the end of sale

**Returns:** prints results to console

2. **Function Name:**

ticket\_logic

**Description:**

Implements the sales loop. Prompts the current buyer, validates input of 1 to 4 tickets, ensures the request does not exceed inventory count, updates remaining ticket count, and counts buyers until inventory reaches zero.

**Parameters:**

tickets (int) – initial number of tickets available at 20

**Variables:**

buyer\_counter (int) – holds current count for amount of buyers starting at 0 and adding 1 per purchase

Purchase (int) – number of tickets requested by the buyer via user input

**Logical Steps:**

1. Initialize buyer\_counter to 0
2. While tickets >0, prompts the user for purchase
3. Validate input from user as an integer
4. Validate input from user is between 1 and 4, if not re-prompt for buyer
5. Check Purchase is <= to tickets, deduct the purchase number from tickets, add 1 to buyer, and display tickets remaining
6. Check if purchase is > to tickets, re-prompt for user to choose less than current ticket count
7. Tickets = 0, exit loop and return to buyer\_counter

**Returns:**

buyer\_counter (int) – final total number of buyers

3. 2. **Function Name:**

Display\_now

**Description:**

Prints the final “sold out” output message and total buyers count

**Parameters:**

buyer\_counter (int) –total number of buyers to display

**Variables:**

(none)

**Logical Steps:**

1. Print message indicating tickets are sold out
2. Print the total number of buyers

**Returns:**

(none)

**Logical Steps:**

1. main() when the file runs
2. main() calls ticket\_logic(20) for the sale loop
3. Ticket\_logic() process purchases and returns buyer\_counter
4. main() calls display\_now(buyer\_counter) to final print

**Link to your repository:** https://github.com/Bisquick-Sand/COP2373

**Output Screenshot:**

A screenshot of a computer program

AI-generated content may be incorrect.