**Technical Design Document: Programming Exercise 1**

**Name:** Presley Reich

**Date Created:** 8/18/2025

**Program Description:**

This program is a ticket selling system. The event is limited to 20 tickets, and customers can purchase up to 4 tickets. At the end of the program the system will display the sold out display along with the number of a customers.

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

1. **Function Name:** get\_tickets

**Description:** This functions prompts the customer to enter the number of tickets they want to buy.

**Parameters:** tickets\_left\_over (int): the number of tickets still available for sale.

**Variables:** num\_str (str): Stores the user input from the console.

num (int): Stores the numeric conversion of the input if valid.

**Logical Steps:**

1. First start a loop that continues until a person enters a number/integer
2. Asks person for the number of tickets they want to buy
3. Checks if the number is numeric
   1. Converts to integer
   2. Checks if the number is between 1-4 and doesn’t exceed left over tickets
4. If valid then returns number of tickets bought
5. If invalid, it lets person know of the limits, if not numeric it prints an error message

**Returns:** num variable (number of tickets customer wants to purchase)

1. **Function Name:** process\_purchase

**Description:** Processes the customer’s ticket purchase, and updates the variables of the remaining tickets, customers that have purchased tickets, and displays the summary of all the purchases made through the system.

**Parameters:** tickets\_left\_over(int): Current number of tickets still available

Customers\_count(int): Total number of customers who have made purchases so far

**Variables:** tickets\_sold(int): The number of tickets sold to the current customer

**Logical Steps:**

1. call get\_tickets to get number of tickets the customer wants to buy
2. Subtract tickets\_sold from tickets\_left\_over
3. Add 1 to customers\_count
4. Print how many tickets are remaining
5. Return the ticket count and customer count

**Returns:** tickets\_left\_over and customers\_count

**3**. **Function Name:** display\_summary

**Description:** Displays final summary message when all tickets have been sold off.

**Parameters:** customers\_count(int): total number of customers who purchased tickets

**Variables:** N/A

**Logical Steps:**

1. Print a message showing all tickets have been sold
2. Display the total number of customers

**Returns:** N/A

**4. Function Name:** main

**Description:** Starts the entire program of the ticket selling program.

**Parameters:** None

**Variables:** Total\_tickets(int) tickets at the start of sale (20), customers(int): counter for total customers (starts at 0).

**Logical Steps:**

1. Intiaizlie total\_tickets and customers variables
2. Loop until all tickets are sold
3. Call process\_purcahse to handle the customer purchase
4. After all tickets are sold, it calls display\_summary to show the final results

**Returns:** [Describe what the function returns]

3. [Add more functions as needed]

**Logical Steps:**

1.main() is called to start the program.

2. main() initializes variables and enters a loop.

3. process\_purchase() is called for each customer.

4. Inside process\_purchase(), get\_tickets() is called to get validated input.

5. After updating ticket count and customer count, process\_purchase() returns to main().

6. Once all tickets are sold, display\_summary() is called to show the final summary.

Note: I tried something new with the isnumeric string method that we learned from last year instead of using trys/exceptions. I think exceptions make it way easier but isnumeric is much easier to see the looping.

**Link to your repository:** <https://github.com/FlyByWireSystems/COP2373>

**Output Screenshot: (make sure big enough so I can see)**

A screen shot of a computer program

AI-generated content may be incorrect.