**Technical Design Document**

Brayden Fox

05/14/2025

**Description:**

This program sells a pre-defined limited number of cinema tickets, where each buyer can buy up to 4 tickets. The maximum number of tickets available is 20. The program prompts the user for a desired number of tickets, confirms the quantity, and displays the total remaining tickets after the sale. It repeats until all tickets are sold, whereby it displays the total number of buyers.

**Functions (in call order):**

1. **Function Name: `**main()`.

**Description:** contains the main program.

**Parameters:** none.

**Variables:**

* `TOTAL\_NOF\_TICKETS`: constant that contains the total number of tickets available at the start of the program.
* `MAX\_TICKETS\_PER\_BUYER`: constant that contains the purchase limit per buyer.
* `curr\_nof\_tickets`: contains current number of tickets remaining after subsequent purchases.
* `qty\_per\_buyer`: a list containing the quantity purchased by each buyer sequentially; the list length “stores” the number of buyers displayed at the end of the program.
* `curr\_max\_qty`: calculated after each iteration; contains the maximum number of tickets available for purchase, which may be less than `MAX\_TICKETS\_PER\_BUYER` when `curr\_nof\_tickets` is lesser than.
* `curr\_qty`: the current quantity desired by the buyer.

**Stepwise Logic:**

* 1. Initialize the main program constants.
  2. Initialize the main program variables.
  3. Enter the main loop and iterate there until no tickets remain.
  4. Determine the maximum number of tickets allowed to be sold at this iteration.
  5. Enter the input validation loop, which is broken out of when the validation conditions are met.
  6. Collect the desired number of tickets from the user, validate whether they are within the allowed range.
  7. If not, inform the user of the error and collect another input; otherwise, break from the loop and proceed.
  8. Add the desired quantity to `qty\_per\_buyer`, subtract the desired quantity from `curr\_nof\_tickets`, and reiterate.

**Return(s):** none.

1. **Function Name:** `collect\_int\_from\_user()`.

**Description:** prompts andcollects an integer response from the user with input validation.

**Parameters:** none.

**Variables:**

* `qty`: string containing the input from the user.

**Stepwise Logic:**

* Enter an infinite loop which is used for reiterating the input prompt in the case that input validation fails.
* Collect value from the user.
* Determine whether the value contains a string of only digits.
* If false, then inform the user and reiterate the validation loop.
* If true, then break from the loop and return the integer form of the digit string.

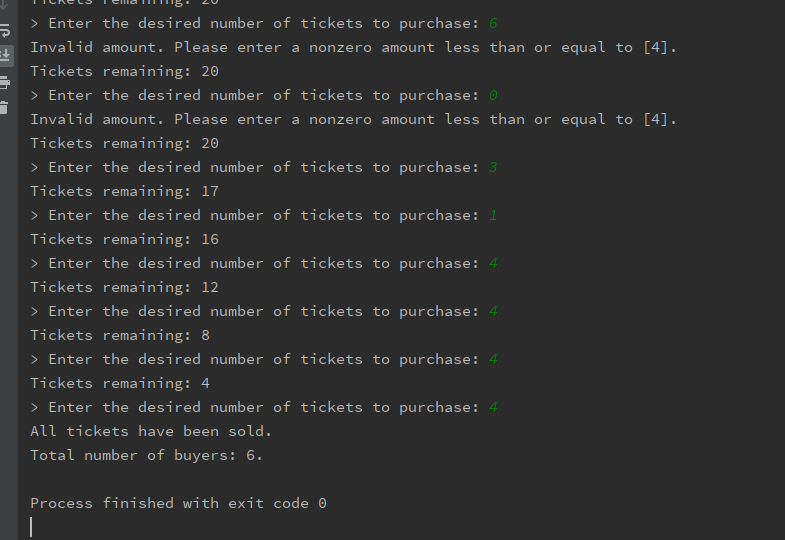
**Return(s):** integer input of the user.

**Logical Structure:**

1. The program functions are defined.
2. The program encounters the `if \_\_name\_\_ == "\_\_main\_\_":` block and calls the main function `main()`.
3. The main program variables are initialized, and the main loop is engaged.
4. The current maximum number of tickets allowed to be sold this iteration is calculated, and execution enters the input validation loop.
5. A quantity is collected from the user via `collect\_int\_from\_user()`and validated against the current maximum quantity allowed to be sold, `curr\_max\_qty`.
6. Upon successful entry, the validation loop is exited and the `qty\_per\_buyer` list and `curr\_nof\_tickets` variables are updated.
7. The main loop continues until all tickets have been sold.
8. After all tickets have been sold, a total is printed, and the program exits.

**Repository:** <https://github.com/fox-2-4/COP2373>

**Output Screenshot(s):**

****