**Programming Assignment: Simple Timer**

**Objective**

Create a countdown timer that takes user input for the duration (in seconds) and displays the time remaining in real-time. The timer should notify the user when the countdown is complete.

**Learning Goals**

By completing this assignment, you will:

1. Practice working with time functions and loops.
2. Learn how to format and display time dynamically.
3. Handle user input and basic input validation.

**Assignment Requirements**

**1. Input**

* Prompt the user to enter the countdown time in seconds.
* Validate the input:
  + Ensure it is a positive integer.
  + Provide feedback if the input is invalid (e.g., "Please enter a valid number greater than 0.").

**2. Timer Functionality**

* Start counting down from the given input.
* Display the time remaining in the format MM:SS (e.g., 02:30 for 2 minutes and 30 seconds).
* Update the display every second.

**3. Completion Notification**

* Notify the user when the timer reaches zero.
  + Example: Print "Time's up!" or play a sound (if possible in your programming environment).

**4. Optional Enhancements (Bonus)**

You can extend the assignment with the following features:

* **Pause and Reset Options:** Allow the user to pause or reset the timer during the countdown.
* **Custom Time Input Format:** Accept input in MM:SS format in addition to seconds.
* **Custom Notification:** Play a sound or display a pop-up message when the countdown is complete.
* **User Interface (Advanced Bonus):** Create a graphical timer display using a GUI library (e.g., tkinter for Python, or equivalent for other languages).

**Example Output**

**Input Prompt:**

Enter the countdown time in seconds: 120

**Timer Output:**

Countdown starts... Time remaining: 02:00 Time remaining: 01:59 Time remaining: 01:58 ... Time remaining: 00:01 Time's up!

**Hints and Tips**

1. **Handling Time Updates:**
   * Use a loop to decrement the timer by 1 second and update the display.
   * Pause the program for a second between updates (e.g., use a time delay function in your chosen language).
2. **Formatting Time:**
   * Convert seconds into minutes and seconds (e.g., using division and modulo operations).
   * Format the time output as MM:SS.
3. **Input Validation:**
   * Ensure the user input is numeric and greater than 0.
   * Use a loop to re-prompt for input if the validation fails.
4. **Structure Your Code:**
   * Consider breaking the task into smaller functions (e.g., one for input validation, one for running the timer).

**Deliverables**

Submit the following:

1. The complete source code of your timer program on github.
2. Notes on any bonus features you implemented.