Decomposition:

1. Design and create a user interface with **buttons for start, stop, and reset**, as well as a display area for showing the elapsed time.
2. Implement a **timer mechanism** that starts when the user clicks the start button, updates the elapsed time continuously, and stops when the user clicks the stop button.
3. **Capture user input** by detecting button clicks and triggering the **corresponding actions**, such as starting, stopping, or resetting the timer.
4. **Continuously update** the displayed elapsed time on the user interface as the timer progresses, ensuring appropriate formatting and presentation.

Pattern Recognition:

1. Recognize the pattern of **starting** and **stopping** a timer to measure elapsed time.
2. Identify the need to continuously **update** **the displayed elapsed time** based on the timer's current value.
3. Observe the pattern of **resetting the timer to zero** and **starting the measurement again**.

Abstraction:

1. Create a stopwatch program that allows users **to measure and display elapsed time**.
2. **Provide** a user interface with **intuitive buttons** for starting, stopping, and resetting the timer.
3. **Display** the **elapsed time** on the user interface, ensuring it is formatted appropriately for easy readability.

Flowchart:

Start

Display Stopwatch Interface

Initialize Elapsed Time to Zero

Display Elapsed Time

While Start Button is not Pressed

Wait for User Interaction

Start Timer

While Stop Button is not Pressed

Update Elapsed Time

Display Elapsed Time

Wait for User Interaction

Stop Timer

If Reset Button is Pressed

Reset Elapsed Time to Zero

Display Elapsed Time

End

Start

While Stop Button is not Pressed

Display Stop Watch Interface

Initialize Elapsed Time to Zero

Update Elapsed Time

Display Elapsed Time

Display Elapsed Time

While Start Button is not Pressed

Wait for User Interaction

Display Elapsed Time

Stop Timer

End

If Reset Button is Pressed

Wait for User Interaction

Start Timer

Reset Elapsed Time to Zero

Algorithm:

1. Display the stopwatch interface.

2. Initialize the elapsed time variable to zero.

3. Display the elapsed time on the stopwatch interface.

4. Wait for the user to press the start button.

5. Start the timer.

6. Enter a loop until the stop button is pressed:

- Update the elapsed time by a small unit (e.g., milliseconds).

- Display the updated elapsed time on the stopwatch interface.

- Wait for the user to press the stop button.

7. Stop the timer.

8. If the user presses the reset button:

- Reset the elapsed time variable to zero.

- Display the elapsed time on the stopwatch interface.

9. End the program.