

Lab 01: Build a console-based stopwatch application

Submission Deadline: Dec 13, 2024

Submission Instruction: Upload to a github repo and submit the repo link on through Google Classroom (link will be provided).

Instructions

Create a console application that can start, stop and reset a stopwatch. Use events to notify the user when specific actions occur (e.g stopwatch started, stopped or reset). Link UI actions (user inputs) with event-handling methods using delegates.

- Start by implementing the Stopwatch class with fields, methods, and events.
- Add event handlers in the main program to listen for events from the stopwatch.
- Build the console UI for interacting with the stopwatch.

Implementation Steps:

Create the Stopwatch Class

- Fields: TimeElapsed (e.g., an integer or TimeSpan), bool IsRunning.
- Methods: Start(), Stop(), Reset(), Tick() (increments time).
- Events: OnStarted, OnStopped, OnReset.
- Use a delegate like public delegate void StopwatchEventHandler(string message);.

Use the event keyword to declare events in the Stopwatch class

- Trigger events within Start(), Stop(), and Reset() methods.

Build the Console UI

- Use a loop to simulate the stopwatch ticking and accept user inputs:
 - Press S to start the stopwatch.
 - Press T to stop it.
 - Press R to reset it.
 - Display time updates in real time (e.g., using Thread.Sleep(1000)).

Link Events to UI Handlers

- Subscribe to OnStarted, OnStopped, and OnReset events in the main program.
- Print messages like "Stopwatch Started!" or "Stopwatch Reset!" when the events are triggered.