1. The Basic Alarm (Define and Invoke)

Goal: Create a class that notifies others when something happens.

- **Task:** Create an AlarmClock class.
- Define a delegate and an event named AlarmRaised.
- Create a method StartAlarm() that triggers the event.
- In your Main method, subscribe to the alarm with a message like "Wake up!" and call StartAlarm().

2. The Video Encoder (Custom EventArgs)

Goal: Pass specific data through an event.

- **Task:** Create a class VideoEncoder and a custom class VideoEventArgs (inheriting from EventArgs).
- VideoEventArgs should contain a Title property.
- The VideoEncoder should have an event VideoEncoded that uses the EventHandler<VideoEventArgs> pattern.
- When a video is "encoded," the event should fire and send the title of the video to all subscribers.

3. The Newsletter Service (Unsubscribe)

Goal: Manage subscriptions dynamically to prevent memory leaks.

- **Task:** Create a Newsletter publisher and a Subscriber class.
- The Subscriber should have a method that prints "Received newsletter."
- In your main logic:
    1. Subscribe two different subscriber objects.
    2. Fire the event (both should receive it).
    3. **Unsubscribe** one of the subscribers.
    4. Fire the event again (only one should receive it).

4. Temperature Monitor (Update UI Logic)

Goal: Simulate how events drive UI updates without the core logic knowing about the UI.

- **Task:** Create a Thermostat class with a CurrentTemperature property.
- Every time the temperature changes, fire a TemperatureChanged event.
- Create a "Display" class that subscribes to this event.
- When the event fires, the Display class should print: *"UI Update: The screen now shows 25°C"*.

5. The Battle System (Decouple Logic)

Goal: Use events to make two classes interact without knowing each other exists.

- **Task:** Create a Player class and a ScoreTracker class.
- The Player should have an event EnemyDefeated.
- The ScoreTracker should handle the logic for incrementing a score.
- **The Rule:** The Player class must **never** have a reference to the ScoreTracker.
- Link them in a separate GameManager or Main method so that when the player defeats an enemy, the score increases automatically.