Chat Room V2

Wednesday, November 5, 2025 7:52 PM

⚙️ Example 2 — Chat Room V2 (Filtered Subscriptions)

🎯 Goal

Upgrade your Chat Room to support **message filtering and smarter subscriptions**, using what you just learned about **RAII + weak_ptr + safe snapshot**.

🧩 Scenario

Your ChatRoom (subject) still manages users (observers), but each user can choose **which messages they want to receive** — using a filter predicate.
Example:

- Alice subscribes to *all* messages.
- Bob subscribes only to messages containing "urgent".
- Carol subscribes only to messages *not* from "SpamBot".

🧠 Acceptance Test

1. Create std::shared_ptr<ChatRoom> room;
2. Create observers:
   - UserDisplay("Alice")
   - KeywordAlert("Bob", "urgent")
   - BlockSender("Carol", "SpamBot")
3. Subscribe each observer with the proper filter:

   ```
   room->subscribe(alice);
   room->subscribe(bob, [](auto& sender, auto& msg){ return msg.find("urgent") != std::string::npos; });
   room->subscribe(carol, [](auto& sender, auto& msg){ return sender != "SpamBot"; });
   ```
4. Send a few messages:
   - "Alice", "hello everyone"
   - "SpamBot", "urgent update!"
   - "Alice", "urgent meeting at 3PM"
5. Expected:
   - Alice sees all.
   - Bob sees only “urgent” ones.
   - Carol ignores SpamBot.
6. Test RAII:
   - Let one Subscription go out of scope → that observer stops receiving.
   - Call removeAll(); new messages produce no output.