■ 10-Hour Order Book Sprint Plan

Hour 0–1: Warm-up & Review

- Re-read current code (`OrderBook`, `Handle`, `Fill`).
- Sketch out data flow: add_limit → match → fill → book update → cancel.
- Write TODO comments where missing pieces are obvious (unit tests, order ID generator, depth utils).

Hour 1–3: Testing Foundation

- Set up Catch2 or GoogleTest in your CMake.
- Write 3 basic unit tests:
  - Add → check resting order in correct side.
  - Cross fill → two fills at correct px/qty.
  - Cancel → remove order and empty level cleanup.
- Confirm tests run green with ctest.

Hour 3–4: Order ID Generator

- Implement a simple monotonic OrderIdGenerator (atomic counter).
- Add to OrderBook so every accepted order gets a unique ID if user doesn’t provide.
- Add test: ensure IDs increment, no duplicates.

Hour 4–6: Depth & Best Price Utilities

- Implement best_bid(), best_ask(), and top_of_book(int depth) returning price/qty snapshots.
- Add tests:
  - After multiple adds/cancels, best_bid/ask update correctly.
  - Depth sums all FIFO orders per level.

Hour 6–7: More Unit Tests

- Edge cases:
  - Partial fill leaves remainder resting.
  - Cross large order through multiple price levels.
  - Cancel nonexistent order returns false.
- Aim: ~15 passing tests covering >80% of code.

Hour 7–8: PnL & Fill Logger

- Add a simple PnL tracker:
  - On each Fill, update cash and inventory.
  - Compute mark-to-market given mid.
- Add a CSV logger for fills (fills.csv: id, side, px, qty, ts).

Hour 8–9: Strategy Skeleton

- Implement a trivial strategy driver:
• Read in mid-price series (or synthetic).
• Quote symmetric bid/ask using add_limit.
• Cancel/refresh old orders every N ticks.
- Confirm fills + PnL update.

Hour 9–10: Wrap-up & Stretch

- Polish: fix typos, run clang-tidy.
- Write README: how to run tests, what’s implemented.
- If time left: stub IOC order support (execute only, no rest).

■ Deliverables by End of Day

- Passing test suite (~15 tests).
- Stable add_limit + cancel with ID generation.
- best_bid/ask + depth snapshot API.
- Fill logger + simple PnL calc.
- Tiny driver strategy that produces fills.
- Clean README + commit history.