Advanced Rust 2026 - Lab 5: Unsafe Boundaries and Sound Wrappers

Lukáš Hozda

Spring 2026

Lab Goals

1. Practice writing minimal unsafe cores.
2. Document invariants explicitly.
3. Prepare for homework pair A5.

Time Plan (90 min)

1. 15 min - unsafe contract checklist
2. 25 min - exercise 1 (safe wrapper around raw memory)
3. 25 min - exercise 2 (sound slice exposure)
4. 15 min - exercise 3 (negative tests / misuse attempts)
5. 10 min - review

Exercise 1: Bounded Buffer Wrapper

Implement wrapper over fixed-capacity raw buffer.

Requirements:

1. Push/pop API is safe.
2. Capacity invariant always maintained.
3. Unsafe block is tiny and justified.

Exercise 2: Borrowed View API

Expose immutable slice view and mutable iterator safely.

Checklist:

1. No aliasing violations.
2. No use-after-free.
3. Lifetime annotations match ownership semantics.

Exercise 3: Misuse Scenarios

Write tests attempting to break invariants:

1. pop from empty
2. push beyond capacity

3. stale index assumptions

Document why safe API prevents UB.