University of Toronto, Department of Electrical and Computer Engineering
ECE 1724-F1 — Performant Software Systems with Rust

Quiz 5 — Sample Quiz 3 — Answer Key

Wednesday, November 25, 2026

1. Ⓑ **Callers must respect the declared lifetime relationship.**

The signature uses the same lifetime for both inputs and the output. The returned borrow is therefore constrained by inner, even though this implementation always selects left. The later println would require that borrow to outlive inner. (Book 10.3.)

2. Ⓐ **Method lifetime elision gives self special treatment.**

When a method has a self reference, lifetime elision assigns that reference lifetime to the returned reference. The extra fallback reference does not make this signature ambiguous; the program prints ready. (Book 10.3.)

3. Ⓒ **A closure can move its captured value out once.**

move transfers saved into the closure. Calling the closure returns that owned String, consuming its captured value. unwrap_or_else accepts a closure that can be called once, so the None case successfully produces cached. (Book 13.1.)

4. Ⓓ **sum consumes the iterator, not the borrowed vector.**

map produces 3, 5, and 7, whose sum is 15. iter borrows values, so consuming that iterator leaves the original vector available and still containing three elements. (Book 13.2.)

5. Ⓑ **Moving a Box transfers ownership of its allocation.**

The assignment moves the Box rather than cloning or sharing its contents. second owns the same heap allocation and prints log. first cannot be used after the move. (Book 15.1.)

Sources and adaptations: Original teaching variations based on the Rust standard library public documentation and the assigned book sections. No internal implementation details are tested.
Q1 source: Original teaching example contrasting a declared lifetime contract with a selected function branch.
Q2 source: Original method example applying the Book self-reference elision rule; no standard-library internals are reproduced.
Q3 source: Original ownership example using the one-call fallback closure accepted by Option.
Q4 source: Original example combining borrowed iteration, map, and the standard sum consumer.
Q5 source: Original ownership-transfer example using Box::new and the documented move behavior of boxes.

Reading: *The Rust Programming Language*, 10.3: Lifetimes, 13.1: Closures, 13.2: Iterators, and 15.1: Box.