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

Quiz 5 — Instructor Answer Key

Wednesday, November 25, 2026

1. Ⓐ **The returned reference is tied only to text.**

The signature ties the result to text through 'a. The note parameter has a separate elided lifetime and does not constrain the result. text is still alive when result is printed; trim returns its ready slice without surrounding spaces. (Book 10.3.)

2. Ⓒ **A lifetime annotation does not extend the owner's lifetime.**

View stores a borrowed string slice; it does not own or clone message. message is dropped at the inner closing brace, but the stored reference is used afterward. The annotation describes the borrowing relationship and cannot keep message alive. (Book 10.3.)

3. Ⓑ **Only the missing value invokes the fallback closure.**

Some(5) returns 5 without executing its closure. None invokes the second closure, which mutably borrows calls, increments it once, and returns 9. The two closure borrows occur in separate calls. (Book 13.1.)

4. Ⓓ **The iterator continues after the element consumed by next.**

next returns a reference to 2 and advances the iterator once. map then doubles only the remaining values, 4 and 6. sum consumes those mapped values, producing 8 + 12 = 20. The underlying vector stays alive while first is used. (Book 13.2.)

5. Ⓒ **A box breaks recursive inline storage.**

Each Node stores a fixed-size owning pointer rather than another Chain inline. The pointee lives on the heap, so the compiler can determine the enum's size without expanding the recursive definition indefinitely. Boxes do not restrict the chain to one node or make it static. (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 two-input wrapper around str::trim, whose returned slice borrows the receiver.
Q2 source: Original teaching example of the borrowed-field pattern described in Book 10.3.
Q3 source: Teaching variation of Option::unwrap_or_else using a counter to expose lazy fallback evaluation.
Q4 source: Original combination of the standard Iterator::next, map, and sum examples.
Q5 source: Renamed and shortened teaching variation of the recursive List example in alloc/src/boxed.rs documentation.

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