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

Quiz 3 — Answer Key to Sample Questions

Wednesday, October 14, 2026

1. Ⓒ **Struct update copies integer fields.**

The update copies the `u32` height into `second`. Changing `second.height` leaves `first.height` unchanged, so the output is `7 2`. (Book 5.1.)

2. Ⓓ **Derived debug formatting.**

`#[derive(Debug)]` enables the `{:?}` formatter for this struct. It does not enable the default `{}` formatter. `d` is not a Rust formatting trait, and `println!` requires a format string. (Book 5.2.)

3. Ⓐ **Methods can borrow their receiver.**

The `&self` receiver immutably borrows `panel`; it neither moves nor changes it. The area is `15`, and the width remains accessible as `3`. (Book 5.3.)

4. Ⓑ **A mutable receiver needs a mutable binding.**

Calling `increment` requires a mutable borrow of `counter`. Declaring it with `let mut counter` allows the call and would produce `5`. (Book 5.3.)

5. Ⓒ **Associated functions use the type name.**

`new` has no `self` parameter. It is called with `Badge::new(8)`, returning a struct whose `number` field is `8`. (Book 5.3.)

6. Ⓐ **Enum variants have their own data shapes.**

`Move` has named fields, so its construction uses braces and field names. `Stop` has no payload; `Message` holds one unnamed `String` value. (Book 6.1.)

7. Ⓓ **Matching an optional value.**

`Some(4)` selects the first arm and binds `points` to `4`. The arm evaluates to `10`. `Some` and `None` are available without the `Option::` prefix. (Book 6.1–6.2.)

8. Ⓑ **Match must be exhaustive.**

The function must handle every possible `Signal`, even though this call passes `Green`. Adding a `Yellow` arm or a catch-all arm makes the match exhaustive. (Book 6.2.)

9. Ⓒ **An unmatched if-let pattern takes the else branch.**

`None` does not match `Some(value)`, so the program prints `unlimited`. An `if let` does not require an exhaustive list of patterns. (Book 6.3.)

10. Ⓐ **Let-else separates success from early return.**

For `None`, the else block returns `0` immediately. For `Some(5)`, the binding succeeds and `number + 1` evaluates to `6`. (Book 6.3.)

Sources: *The Rust Programming Language*, 5.1: Defining Structs, 5.2: An Example Program Using Structs, 5.3: Method Syntax, 6.1: Defining an Enum, 6.2: Match, and 6.3: If Let and Let Else.