UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 2 — Answer Key

Coverage: Rust Book Chapters 1–6, 8, and 9. Chapter 7 is excluded.

110 minutes **40 questions** **100 marks**

Section A: 2 marks per question. Section B: 3 marks per question. References below identify sections of *The Rust Programming Language*.

Question 1: (A) 2 marks

cargo build --release selects Cargo's release build profile, which enables optimizations. The resulting executable is placed under target/release. A normal cargo build uses the development profile instead. *Book: 1.3.*

Question 2: (D) 2 marks

cargo run handles both building and executing the binary. cargo build builds the package but does not run it. This distinction lets you choose whether to execute the program immediately after compilation. *Book: 1.3.*

Question 3: (B) 2 marks

parse does not remove surrounding whitespace from numeric input. Without trim, this input produces an Err. The match handles the error by printing invalid instead of panicking. *Book: 2.*

Question 4: (D) 2 marks

cmp compares measured with target and returns Ordering::Less because 8 is smaller than 10. The match executes only the Less arm, which prints low. *Book: 2.*

Question 5: (D) 2 marks

mut permits changing a binding's value, but the new value must have the same type. Assigning a string slice to this integer binding fails. A new let binding could shadow it with a different type. *Book: 3.1.*

Question 6: (D) 2 marks

The array expression [6; 3] creates three elements, each containing 6. Therefore the element at index one is 6 and the array length is 3. The second number is a count, not an element. *Book: 3.2.*

Question 7: (A) 2 marks

The semicolon turns value + 3 into a statement rather than the function's return expression. The function body therefore returns the unit value, which does not match its declared i32 return type. *Book: 3.3.*

Question 8: (B) 2 marks

The String first moves from station into saved, then from saved into final_name. Each move transfers ownership. The code uses only the final owner afterward, so it compiles and prints north. *Book: 4.1.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 2 — Answer Key (continued)

Coverage: Rust Book Chapters 1–6, 8, and 9. Chapter 7 is excluded.

Question 9: (D) 2 marks

Passing an integer to increase copies its value into the parameter. The function modifies and prints its local copy, producing 8. The caller's separate count remains 7 and is still available. *Book: 4.1.*

Question 10: (D) 2 marks

The open-ended slice begins at byte index five and extends to the end. The character at index five is 4, followed by 2. The hyphen is at index four and is excluded. *Book: 4.3.*

Question 11: (A) 2 marks

Field initialization shorthand uses the binding with the same name as each field. Reading { value, valid } therefore sets the fields to 12 and false, just as explicit value: value and valid: valid would. *Book: 5.1.*

Question 12: (C) 2 marks

Deriving Debug supports the debug formatting placeholder {:?}. This println uses the normal display placeholder {}, which requires Display. Deriving Debug does not provide Display, so compilation fails. *Book: 5.2.*

Question 13: (B) 2 marks

weather holds the Rain variant with payload 3. The match binds that payload to mm and evaluates mm * 2, producing 6. The Dry variant's arm is not selected. *Book: 6.1–6.2.*

Question 14: (C) 2 marks

A match on Option must handle both Some and None, even when this particular value was initialized with Some. The missing None case makes the match non-exhaustive, causing a compilation error. *Book: 6.2.*

Question 15: (C) 2 marks

The tuple contains the first and third array elements, 3 and 9. Destructuring binds those values to first and last. Their sum is 12, and the array length is 3. *Book: 3.2.*

Question 16: (A) 2 marks

The value Some(4) matches the Some(value) pattern, binding value to 4. The block adds that value to the initial count of 1, so the final println prints 5. *Book: 6.3.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 2 — Answer Key (continued)

Coverage: Rust Book Chapters 1–6, 8, and 9. Chapter 7 is excluded.

Question 17: (D) 2 marks

The vector has two elements, so its valid indices are zero and one. Indexing at two performs a bounds check and panics. Unlike get, indexing does not return None for an invalid index. *Book: 8.1; 9.1.*

Question 18: (D) 2 marks

String::len reports bytes, not the number of characters. The initial character occupies two bytes, and a, s, and t occupy one each. The total byte length is therefore 5. *Book: 8.2.*

Question 19: (C) 2 marks

or_insert supplies a value only when the entry does not already exist. rain already maps to 2, so that value remains unchanged. get finds the existing entry; unwrap and dereferencing retrieve the value 2. *Book: 8.3.*

Question 20: (B) 2 marks

Parsing rain produces Err. Calling expect on that error panics, so count is never assigned a parsed number and println is not reached. The expect message is part of the panic diagnostic, not normal printed output. *Book: 9.2.*

Question 21: (A) 3 marks

The first two lines parse as 8 and 3, contributing 8 and 1. The invalid third line increments skipped and breaks out of the loop. The final line is never processed, leaving total 9 and skipped 1. *Book: 2; 3.5; 9.2.*

Question 22: (A) 3 marks

The shadowed levels are 3, 6, 2, and 5. The first three iterations add 1, 6, and 2, making total 9. The break then exits with used equal to 3. *Book: 3.1; 3.5.*

Question 23: (C) 3 marks

backup is an independent clone made before station changes. tag takes ownership of the modified station and returns bay-2-ok. Moving backup into display preserves its original bay contents, then appending ! produces bay!. *Book: 4.1.*

Question 24: (C) 3 marks

The last use of view is the first println, so its immutable borrow ends before add_marker needs a mutable borrow. The marker changes dock to dock!, increasing the byte length from four to five. *Book: 4.2.*

Coverage: Rust Book Chapters 1–6, 8, and 9. Chapter 7 is excluded.

Question 25: (C) 3 marks

The loop finds the space at byte three, making first equal to sun. String::from creates an owned copy. After the slice's last use, report can be cleared and changed without changing saved. *Book: 4.3; 8.2.*

Question 26: (D) 3 marks

revised explicitly supplies its own destination and boxes, so struct update copies only urgent, a bool. No String moves out of original. Its boxes remains 4 and its destination remains west, while revised.boxes is 7. *Book: 5.1; 4.1.*

Question 27: (B) 3 marks

new constructs a gauge at 5. Raising it by four produces 9, so above(8) returns true and stores that boolean. The later change lowers the gauge to 7 without changing the previously computed boolean. *Book: 5.3.*

Question 28: (B) 3 marks

Warm(3) contributes 3 times 2, or 6. Rain(4) contributes 4 + 1, or 5. Clear contributes zero. Each event selects its own match arm, producing a total of 11. *Book: 6.1–6.2.*

Question 29: (B) 3 marks

adjust(Some(4)) returns Some(6), while adjust(Some(0)) returns None because zero is not positive. The later matches extract 6 from the first result and choose the fallback 10 for the second. *Book: 6.2.*

Question 30: (A) 3 marks

Some(7) passes the let-else pattern and adds seven. None takes the else block, adds one, and returns. Some(3) reaches the final else branch and adds two, giving a total of ten. *Book: 6.3; 4.2.*

Question 31: (B) 3 marks

width borrows record without consuming it. into_label takes self by value and therefore consumes record. The later call to record.width() uses a moved value, causing compilation to fail before any output. *Book: 5.3; 4.1.*

Question 32: (B) 3 marks

The updates leave current at 5, 3, 0, and 7 after successive matches. Each of those values contributes to total, giving 15. Reset clears current at that point but does not reset total. *Book: 6.1–6.2; 3.5.*

Coverage: Rust Book Chapters 1–6, 8, and 9. Chapter 7 is excluded.

Question 33: (C) 3 marks

The vector starts with 2. Successful parses append 3 and 5 and increment accepted; the invalid string changes neither. The final vector is [2, 3, 5], whose sum is 10, length is 3, and accepted count is 2. *Book: 8.1; 9.2.*

Question 34: (C) 3 marks

The mutable loop produces [3, 5, 7]. first is 3, so push adds 5. The second loop contributes 1 for 3 and the full values 5, 7, and 5, totaling 18 across four elements. *Book: 8.1.*

Question 35: (B) 3 marks

The vector holds values of the same enum type with different payloads. Loaded and Unloaded change boxes by +8, -3, and +2, giving 7. The single note contributes its four-byte length, giving notes 4. *Book: 8.1; 6.2.*

Question 36: (C) 3 marks

format! borrows both strings and creates east:gate before area changes. Concatenation consumes area but borrows suffix, producing east!-gate with ten bytes. short is a clone, so appending 2 leaves suffix unchanged. *Book: 8.2; 4.1.*

Question 37: (A) 3 marks

The counting loop yields rain 2, sun 1, and wind 1. or_insert leaves existing values unchanged, so sun stays 1. Wind increases to 3, and insert replaces rain's value with 7. *Book: 8.3.*

Question 38: (D) 3 marks

The first call parses both strings and returns Ok(12). The second call encounters an error on its first parse; ? returns that error immediately. main converts the error to -1 using its match. *Book: 9.2.*

Question 39: (A) 3 marks

The first reading succeeds with 6 and updates last. The invalid reading uses that current last value, also 6. The final reading succeeds with 2. Thus total is 6 + 6 + 2 = 14 and last is 2. *Book: 9.2.*

Question 40: (A) 3 marks

The successful first result prints loaded 4. Parsing bad returns Err without panicking, so checked is printed next. expect then panics on that Err, preventing both remaining println calls from executing. *Book: 9.1–9.3.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 2 — Coverage and Timing

Coverage: Rust Book Chapters 1–6, 8, and 9. Chapter 7 is excluded.

Scoring and suggested timing

Part	Questions	Marks	Time
Easy	1–20	40	40 minutes
Longer code reading	21–40	60	about 67 minutes
Review	All	—	about 3 minutes
Total	**40**	**100**	**110 minutes**

The pacing follows five easy questions per ten minutes and three harder questions per ten minutes. Harder questions combine familiar concepts in longer programs; they do not depend on advanced Rust topics.

Chapter coverage (primary and supporting topics)

Chapter	Questions
1: Getting started	1–2
2: Guessing game fundamentals	3–4, 21
3: Common programming concepts	5–7, 15, 21–22, 32
4: Ownership	8–10, 23–26, 30–31, 36
5: Structs	11–12, 26–27, 31
6: Enums and pattern matching	13–14, 16, 28–30, 32, 35
8: Common collections	17–19, 25, 33–37
9: Error handling	17, 20–21, 33, 38–40

Source: official Rust Book

Chapter 1 Chapter 2 Chapter 3 Chapter 4 Chapter 5 Chapter 6 Chapter 8 Chapter 9

All executable examples were checked with Rust 2024. Output choices separated by *then* or semicolons describe successive output lines.