UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 1 — Answer Key

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

110 minutes **40 multiple-choice questions** **100 marks**

Section A: 20 questions × 2 marks = 40. Section B: 20 questions × 3 marks = 60.

Question 1: (B) 2 marks

Cargo check performs compilation checks without generating the executable. Cargo build produces build artifacts; cargo run builds and runs the program; cargo new creates a project. *Book: 1.3.*

Question 2: (D) 2 marks

Cargo places the default binary source and its main function in src/main.rs. Cargo.toml contains package configuration and dependencies, while target holds generated build artifacts. A library normally uses src/lib.rs. *Book: 1.3.*

Question 3: (A) 2 marks

Trimming removes the leading space and trailing newline before parsing. The type annotation selects an unsigned integer, parsing succeeds, and the final expression adds one to 42. *Book: 2.*

Question 4: (C) 2 marks

read_line borrows the mutable String and appends the entered bytes, including the newline. It returns a Result describing success or failure; expect handles that Result. The String remains owned by the caller. *Book: 2.*

Question 5: (D) 2 marks

Each let creates a new binding that shadows the previous binding. The intermediate value is 6 and the final value is 18. Shadowing does not require the original binding to be mutable. *Book: 3.1.*

Question 6: (B) 2 marks

The final expression has no semicolon, so its value is returned from charge. Substituting 3 for days gives 3 times 2 plus 1, which is 7. *Book: 3.3.*

Question 7: (A) 2 marks

Array indexing starts at zero, so desks[1] is 5 and desks[2] is 8. Tuple fields also start at zero; the two selected values print in their original order. *Book: 3.2.*

Question 8: (C) 2 marks

Assigning the String to assigned transfers ownership. String does not implement Copy, so the old ticket binding cannot be used in println afterward. The compiler rejects the program before it runs. *Book: 4.1.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 1 — Answer Key (continued)

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

Question 9: (B) 2 marks

Cloning creates an independently owned String with the same initial text. Appending to original changes only that String, so saved still contains loan when the two values are printed. *Book: 4.1.*

Question 10: (D) 2 marks

The slice begins at byte zero and excludes byte four. These ASCII characters each occupy one byte, so the selected bytes spell Rust without the following space. *Book: 4.3.*

Question 11: (A) 2 marks

Declaring the entire loan binding mutable permits assigning to its fields. Rust does not put mut on individual field declarations; increasing days from 4 by 2 produces 6. *Book: 5.1.*

Question 12: (C) 2 marks

The method borrows the desk immutably through &self, so the binding need not be mutable. Its comparison checks whether two is greater than zero and returns true. *Book: 5.3.*

Question 13: (D) 2 marks

Some(7) matches the Some arm and binds 7 to n. That arm evaluates to 8, which becomes number. The None arm is present for completeness but is not selected. *Book: 6.1–6.2.*

Question 14: (A) 2 marks

The if let body executes only when note contains Some. Here note is None, so the body is skipped and count retains its original value of 2. *Book: 6.3.*

Question 15: (B) 2 marks

The array binding is mutable, so its second element can be assigned a new value. previous holds a copied integer, 4; changing the array element to 7 does not change that copy. *Book: 3.1–3.2.*

Question 16: (C) 2 marks

The if let matches Some(3), binds its integer payload, and assigns Some(5) to the mutable binding. The following match therefore selects its Some arm and prints the updated value, 5. *Book: 6.2–6.3.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 1 — Answer Key (continued)

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

Question 17: (A) 2 marks

Pushing 6 adds a third element to the vector, so its length becomes 3. The element at index 1 is still the original second element, 4. *Book: 8.1.*

Question 18: (D) 2 marks

The push_str call borrows suffix and appends its text to label. It does not take ownership of suffix, which remains available for printing after the separator. *Book: 8.2.*

Question 19: (C) 2 marks

Inserting a value for an existing key replaces that key's previous value. The second insertion changes open from 3 to 5; it does not add the two values together. *Book: 8.3.*

Question 20: (B) 2 marks

Parsing this nonnumeric text returns an Err value. The match handles that error by printing retry. A recoverable error does not itself panic; the caller chooses how to handle it. *Book: 9.2.*

Question 21: (C) 3 marks

Trimming makes the spaced inputs valid. The failed parse increments rejected and skips that iteration. Values 4 and 7 contribute themselves, while 2 contributes 1, giving total 12 and one rejected entry. *Book: 2, 3.5, 9.2.*

Question 22: (A) 3 marks

The first amount becomes 3 and adds 2. The second becomes 6 and adds 6, bringing score to 8. The break ends the loop immediately, so only two deliveries were visited. *Book: 3.1, 3.5.*

Question 23: (D) 3 marks

The original String moves into stamp and returns with an exclamation mark. Its clone remains independent and later becomes archive. Width only borrows updated and returns its five-byte length; no moved binding is reused. *Book: 4.1–4.2.*

Question 24: (B) 3 marks

The last use of view occurs before attach, so its immutable borrow no longer conflicts with the mutable borrow. Appending six ASCII bytes changes the length from 4 to 10, and current observes the updated text. *Book: 4.2.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 1 — Answer Key (continued)

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

Question 25: (A) 3 marks

The first space is at index 6, producing return. Skipping that word and its space leaves blue book, which is nine bytes long. Its first word is blue; all slices borrow immutably. *Book: 4.3, 8.2.*

Question 26: (D) 3 marks

Struct update transfers the String title into second because that field is not supplied explicitly. The integer days and boolean renewed are Copy values. Accessing first.days is valid, but the later first.title access prevents compilation. *Book: 5.1, 4.1.*

Question 27: (B) 3 marks

Desk::new constructs a desk with eight waiting requests. The first mutable method call serves three and leaves five. The second call serves only those five despite its limit of seven, leaving zero. *Book: 5.3.*

Question 28: (C) 3 marks

The match selects an operation from each enum variant. Borrow subtracts three, Return adds two, and Closed adds zero. Starting from ten therefore leaves nine; enum variants may carry different kinds of data. *Book: 6.1–6.2.*

Question 29: (D) 3 marks

The three calls transform Some(4) into Some(5), then None, then Some(1). The last transition uses the None arm rather than stopping the computation. The final match therefore prints 1. *Book: 6.1–6.2.*

Question 30: (B) 3 marks

Some(2) contributes two, Some(5) contributes four, and Some(1) contributes one. None increments missing to one. Because an arrival was missing, the final bonus is one, bringing total to eight. *Book: 6.3, 3.5.*

Question 31: (C) 3 marks

estimate temporarily borrows ticket and returns seven. That borrow is finished before finish takes ownership of ticket and returns its completion message. The final print uses the separate estimate and message values, not ticket. *Book: 5.3, 4.1–4.2.*

Question 32: (A) 3 marks

Adding four, serving one, and adding two leaves five waiting and one served. Reset clears waiting only. The next Serve finds no waiting requests, so served stays one; the final Add leaves three waiting. *Book: 6.1–6.2, 3.5.*

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

Question 33: (B) 3 marks

The successful inputs append four and six to accepted, while the failed parse increments errors to one. Appending that error count makes the vector [4, 6, 1], with three elements and a sum of eleven. *Book: 8.1, 9.2.*

Question 34: (D) 3 marks

The mutable loop changes the vector to [4, 4, 3]. Pushing five produces [4, 4, 3, 5], whose sum is sixteen. The mutable iteration has finished before the later immutable iteration begins. *Book: 8.1.*

Question 35: (C) 3 marks

All vector elements have the same enum type despite using different variants. Each Open ticket adds its minutes plus one: four, seven, and three. The Closed ticket increments only closed, giving fourteen and one. *Book: 8.1, 6.2.*

Question 36: (A) 3 marks

format! creates Lab-7 without taking ownership of its inputs. The later addition consumes room but borrows number, yielding Lab!7 with length five. The log uses the earlier saved text and appends a plus sign. *Book: 8.2, 4.1.*

Question 37: (D) 3 marks

Book is counted three times and desk grows from two to three. The existing book entry ignores the fallback ten, then gains two. Printer is absent, so the match supplies four. No output depends on map iteration order. *Book: 8.3.*

Question 38: (C) 3 marks

The first parse succeeds with six. read_second returns Err containing second, and the question mark immediately returns that error from combined. The caller handles it with the Err arm, printing error second without panicking. *Book: 9.2.*

Question 39: (B) 3 marks

The helper treats both zero and a parse failure as errors. The caller recovers from each by adding one and counting a failure. Successful inputs add three and five, producing total ten and two failures. *Book: 9.2–9.3.*

Question 40: (A) 3 marks

The first Result contains four and is handled successfully. The second parse returns an error but does not panic, so checked prints. Calling expect on that Err then panics, preventing both remaining println calls. *Book: 9.1–9.3.*

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE 1724-F1 — Performant Software Systems with Rust

Sample Midterm 1 — Coverage and Timing

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

Scoring and suggested pacing

Section A (Q1–20): 40 marks, about 40 minutes. At five easy questions per ten minutes, each easy question averages two minutes.

Section B (Q21–40): 60 marks, about 66 minutes 40 seconds. At three harder questions per ten minutes, each harder question averages three minutes 20 seconds.

Total working time: 106 minutes 40 seconds, leaving 3 minutes 20 seconds for review. Harder questions use longer code with familiar concepts from Chapters 1–6, 8, and 9.

Chapter coverage (questions may span chapters)

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

Source: The Rust Programming Language

The question scope follows the official Rust Book. Section references in each answer identify the relevant explanation. Chapter links:

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