CERTLABZ CERTIFICATION BOARD

Certified Scala Programmer (CSLP)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working Scala programmer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: Scala Fundamentals and Syntax

DATA

Emma Wilson maps each Scala type to an example literal, with one example missing.

Type	Example
Int	42
Boolean	(to choose)
String	hello
Double	3.14

Table 1.1 Types and example literals.

Which literal fits Boolean?

A. true
B. "yes"
C. 1
D. 0

Correct answer: A

Boolean values in Scala are the literals true and false, so true fills the blank. The string "yes" is a String, and the numbers 1 and 0 are Int values that Scala does not treat as Boolean.

Question 2 Domain 2: Types, Collections and Pattern Matching CONFIG

James Carter runs this pattern match on a value typed as Any.

```
val x: Any = 5
val r = x match {
  case i: Int => "int"
  case s: String => "str"
  case _ => "other"
}
```

Listing 2.1 A type-based match.

What value does r hold?

A error
B "str"
C "int"
D "other"

Correct answer: C
The value 5 is an Int, so the first case matches and r holds "int". Pattern matching tests each case in order and stops at the first that fits, so the String and wildcard cases are never reached.

Question 3 Domain 3: Control Flow and Functions CONFIG

Olivia Davis defines a recursive function and calls it once.

```
def fact(n: Int): Int =
  if (n <= 1) 1
  else n * fact(n - 1)

fact(4)
```

Listing 3.1 A recursive factorial.

What does fact(4) return?

A 12
B 24
C 10
D 4

Correct answer: B
The function multiplies 4 by 3 by 2 by 1, which gives 24. Each call reduces n by one until the base case n <= 1 returns 1, and the products unwind to the final result.

Question 4

Domain 4: Object-Oriented and Functional Programming

DATA

Liam Anderson maps each Scala construct to how many instances it creates, with one entry missing.

Construct	Instances
class	Many
object	(to choose)
trait	Mixed in
case class	Many, immutable

Table 4.1 Constructs and instances.

How many instances does an object create?

A None at all

B One per call

C Many copies

D Exactly one

Correct answer: D

An object declaration defines a singleton, so Scala creates exactly one instance of it. A class can be instantiated many times, while a trait is mixed into other types rather than instantiated on its own.

Question 5

Domain 5: Immutability, Higher-Order Functions and the Standard Library

CONFIG

Ava Thompson chains two collection operations over a list of numbers.

```
val nums = List(1, 2, 3, 4)
val out = nums
  .filter(_ % 2 == 0)
  .map(_ * 10)
```

Listing 5.1 A filter then map.

What is out?

A List(20, 40)

B List(10, 30)

C List(2, 4)

D List(10,20,30,40)

Correct answer: A

The filter keeps the even numbers 2 and 4, then map multiplies each by 10 to give List(20, 40). Order matters here, since map runs only on the elements that filter has already kept.

Question 6 Domain 6: Concurrency, Tooling and Best Practices DATA

David Brown maps each type to what it represents, with one meaning missing.

Type	Represents
Future	Async result
Try	(to choose)
Option	Maybe a value
Either	One of two

Table 6.1 Types and meaning.

What does Try represent?

A. An async task
B. A future value
C. Ok or error
D. A parallel run

Correct answer: C
A Try holds either a successful result or the error that was thrown, so it captures ok or error. A Future represents an asynchronous result, while running work in parallel is a separate concern from what Try models.

Question 7 Domain 1: Scala Fundamentals and Syntax

A developer wants a name whose value can never change after it is first assigned. Which keyword should they use?

A. Use var
B. Use lazy
C. Use def
D. Use val

Correct answer: D
A val binds a name to a value that cannot be reassigned, which is what the developer wants. A var can be reassigned, def defines a method, and lazy only delays when a val is first evaluated.

Question 8 Domain 4: Object-Oriented and Functional Programming

A team wants to share the same method behaviour across several unrelated classes without forcing them under one base class. What fits best?

A. A trait
B. A var field

C A companion

D A for loop

Correct answer: A
A trait bundles methods and can be mixed into many unrelated classes, so it shares behaviour without a shared base class. A companion object pairs with one type, and fields or loops do not share behaviour across classes.

Question 9

Domain 5: Immutability, Higher-Order Functions and the Standard Library

Code transforms each element into a list and then flattens the nested lists into one, doing it in two separate steps. Which single function does both at once?

A map

B flatMap

C filter

D foldLeft

Correct answer: B
flatMap applies a function that returns a collection and then flattens the results into one collection, replacing the map then flatten pair. A plain map leaves the results nested, filter only selects, and foldLeft reduces to a single value.

Question 10

Domain 6: Concurrency, Tooling and Best Practices

An app blocks a thread while it waits for a slow network call, which freezes the rest of the work. Which type lets the call run without blocking?

A A while loop

B Option

C Future

D A var

Correct answer: C
A Future runs the call on another thread and completes later, so the main work is not blocked while it waits. A while loop, an Option, or a var do not add concurrency and would still leave the thread blocked.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CSLP-001. https://certlabz.com/certifications/certified-scala-programmer.html