CERTLABZ CERTIFICATION BOARD

Certified Corda Developer (CCRD)

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 Corda developer 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: Corda and Enterprise DLT Foundations

DATA

Sarah Miller compares Corda with a public blockchain, with one Corda cell missing.

Aspect	Corda	Public chain
Data sharing	Need to know	Global broadcast
Membership	(to choose)	Open, anonymous
Consensus	Per transaction	Whole ledger
Identity	Known parties	Pseudonymous

Table 1.1 Corda against a public chain.

What describes Corda membership?

A. Open to anyone
B. Anonymous accounts
C. Permissioned network
D. Open, anonymous joining

Correct answer: C

Corda runs as a permissioned network where every node holds a known, certified identity, unlike open public chains. That controlled membership is what lets parties transact on a need to know basis rather than by global broadcast.

Question 2 — Domain 2: States, Contracts and Transactions — CONFIG

James Carter reviews a Kotlin state class in a CorDapp.

```
class IOUState(
  val amount: Int,
  val lender: Party,
  val borrower: Party
) : ContractState {
  override val participants =
    listOf(lender, borrower)
}
```

Listing 2.1 A Corda state class.

What makes this a valid ContractState?

A. Defines participants
B. Extends a base class
C. Holds only strings
D. Runs a notary sub-flow

Correct answer: A
Every ContractState must expose a participants list naming the parties who keep the state in their vaults and must sign to change it. The state does not extend a base class or run flows, and it can hold any serializable fields.

Question 3 — Domain 3: Flows and Consensus — CONFIG

Emma Wilson traces an issuance flow that ends with a call to FinalityFlow.

```
@InitiatingFlow
@StartableByRPC
class IssueFlow(val amt: Int) :
  FlowLogic<SignedTransaction>() {
  @Suspendable
  override fun call(): SignedTransaction {
    val stx = signInitial(build())
    return subFlow(FinalityFlow(stx))
  }
}
```

Listing 3.1 A flow ending in FinalityFlow.

What does FinalityFlow accomplish here?

A. Compiles the CorDapp
B. Creates a new party
C. Opens an RPC port

D Records with notary

Correct answer: D
FinalityFlow sends the signed transaction to the notary for uniqueness consensus and then records the finalized transaction in the vault of every participant. It is the step that commits the transaction rather than compiling code or opening ports.

Question 4

Domain 4: CorDapp Development

CONFIG

David Brown reviews client code that drives a running node from outside.

```
val client = CordaRPCClient(addr)
val proxy = client.start(
  user, password).proxy
val handle = proxy.startFlowDynamic(
  IssueFlow::class.java, 100)
val out = handle.returnValue.get()
```

Listing 4.1 A CorDapp client call.

What does this client code do?

A Signs a certificate

B Calls a flow by RPC

C Deploys a new node

D Edits the network map

Correct answer: B
CordaRPCClient opens an authenticated RPC connection to a node, and startFlowDynamic launches a flow and returns a handle whose returnValue yields the result. It invokes a flow remotely rather than deploying nodes or editing the map.

Question 5

Domain 5: Nodes, Networks and Notaries

DATA

Olivia Davis maps each notary style to its property, with one property missing.

Notary	Property
Non-validating	Sees only inputs
Validating	(to choose)
Raft cluster	Crash tolerant
BFT cluster	Byzantine tolerant

Table 5.1 Notary styles and properties.

What does a validating notary do?

A Checks full tx

B Ignores contracts

C Only stores hashes

D Skips input states

Correct answer: A
A validating notary sees and re-runs the whole transaction, including contract checks, before it signs, which prevents an invalid spend but exposes transaction data to the notary. A non-validating notary checks only the input states for uniqueness.

Question 6

Domain 6: Security, Testing and Deployment

CONFIG

Liam Anderson reads a test harness that stands up nodes in memory.

```
val network = MockNetwork(
  MockNetworkParameters(cordapps))
val a = network.createNode()
val b = network.createNode()
val f = a.startFlow(IssueFlow(100))
network.runNetwork()
val result = f.get()
```

Listing 6.1 A flow test harness.

What is this code set up to do?

A Deploy to production

B Encrypt the vault

C Test a flow offline

D Open a public port

Correct answer: C
MockNetwork spins up in-memory nodes and runNetwork drives the message passing between them, so a flow can be tested repeatably without any real infrastructure. It is a unit test setup, not a production deployment or an encryption step.

Question 7

Domain 2: States, Contracts and Transactions

A developer wants a state to stop being spendable once it is used, so it no longer appears as an unspent output. What mechanism marks it that way?

A Delete the ledger row

B Encrypt the state

C Rename the class

D Consume the output

Correct answer: D
A Corda state is retired when a transaction consumes it as an input, which marks it historic in the vault while a new output carries the value forward. States are never deleted or edited in place, so the ledger stays an append only history.

Question 8 Domain 4: CorDapp Development

After a CorDapp is live, the team must change a contract rule without breaking states already on the ledger. What supports this safely?

A. Rewrite it in place
B. Contract upgrade flow
C. Drop the old states
D. Fork the whole network

Correct answer: B
Corda provides explicit contract and state upgrade flows so the parties agree to move existing states onto a new contract version while the history stays intact. Rewriting a contract in place or dropping states would break the shared ledger.

Question 9 Domain 5: Nodes, Networks and Notaries

Two running nodes cannot locate each other to transact, though both are on the same network. What component do they rely on to find identities and addresses?

A. The network map
B. A public mempool
C. The mining pool
D. A broadcast bus

Correct answer: A
The network map service publishes each node identity and network address so peers can discover one another and connect. Corda has no mempool or mining, which are constructs of public broadcast blockchains rather than a permissioned network.

Question 10 Domain 6: Security, Testing and Deployment

For a production deployment the team needs notarization to keep working even if one notary node fails. What approach best meets this?

A. Run a single notary
B. Disable the notary
C. Share the key
D. Cluster the notary

Correct answer: D

Running the notary as a fault tolerant cluster, such as a Raft or BFT group, keeps notarization available when a member node fails. A single notary is a single point of failure, and disabling it removes protection against double spends.

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