**What does a State in Corda represent?**

窗体顶端

State of all customer accounts

Global state of the ledger

A fact known by one or more parties

窗体底端

 Required

**States must be shared between at least two participants.**

窗体顶端

True

False

**Fields of any ContractState are stored in a separate database columns when persisted. It is possible to run any kind of SQL-like queries against these states.**

窗体顶端

True

False

窗体底端

 Required

**Imagine a class which implements OwnableState interface: class MyCommand : CommandData, TypeOnlyCommandData()  data class MyState(val partyA : AbstractParty, val partyB : AbstractParty) : OwnableState { override val owner = partyA override val participants = listOf(partyA, partyB) override fun withNewOwner(newOwner : AbstractParty) = CommandAndState(MyCommand(), copy(partyA = newOwner)) } Which participants will store the state after it gets committed to the ledger?**

窗体顶端

partyA

partyB

partyA and partyB

The code will not work as list of participants can't include more than 1 party for ownable states.

窗体底端

 Required

**What field(s) does the ContractState interface define?**

窗体顶端

contract

legalProseReference

participants

owner

窗体底端

 Required

**What interface allows states to start a flow at a certain future point in time?**

窗体顶端

SchedulableState

DelayedState

TimeState

FlowState

窗体底端

 Required

**What type of on-ledger facts should be represented using OwnableState?**

窗体顶端

States that can be split and merged with other states of the same type

States that should be stored in the node's vault

States that have a unique identifier over time

States where only owner is required to sign transactions transferring them to another party

窗体底端

 Required

**Where does each node store their states?**

窗体顶端

File System

Database

Vault

Notary

**What does a Contract in Corda represent?**

窗体顶端

Where each node can create a transaction.

Optional documents attached to a transaction for legal purposes.

There are no Contracts in Corda.

The sets of constraints that govern the evolution of state objects.

窗体底端

 Required

**Which of the following statements about contracts is true?**

窗体顶端

Only one type of contract can be referenced in a transaction

All contracts referenced by a transaction will be used to verify the transaction.

States do not need to reference specific contracts

None of the above

窗体底端

 Required

**It is possible to allow non-deterministic Java libraries, such as DateTime, to be used in Contracts if to fulfill business logic.**

窗体顶端

True

False

**Contracts are included in attachments in Corda.**

窗体顶端

True

False

窗体底端

 Required

**What does a Contract.verify method return?**

窗体顶端

Boolean

Integer

ContractState

Void/Unit

**What does a Transaction in Corda represent?**

窗体顶端

A proposal for all nodes on the network to update their ledgers.

A proposal for a number of participants to update their ledgers.

An order for all nodes on the network to update their ledgers.

An order for a number of participants to update their ledgers.

窗体底端

 Required

**For performance reasons, the number of participants in a transaction is limited to:**

窗体顶端

30

50

100

not limited

窗体底端

 Required

**How are transactions consuming or creating multiple state objects processed?**

窗体顶端

Valid state transitions of a transaction are recorded to the ledger. Invalid state transitions return an error.

Transactions with multiple states must contain states of all the same type.

All state changes must be valid. Any invalid state transition will result in a failed transaction.

None of the above

窗体底端

 Required

**What is the purpose of filtering a transaction?**

窗体顶端

Preventing my node from storing input or output states twice

Finding the outputs of a transaction that are relevant to my node

Hiding certain components of the transaction from users who do not need to see them

Reducing the size of messages sent between nodes

窗体底端

 Required

**A transaction creates \_\_\_\_\_\_ output states.**

窗体顶端

0 or more

1 or more

2 or more

at most 1

窗体底端

 Required

**A transaction cannot be broadcasted to the Corda Network.**

窗体顶端

True

False

窗体底端

 Required

**Contract verify is the only code run when verifying a transaction.**

窗体顶端

True

False

**When instantiating a TransactionBuilder the reference to a Notary is an optional argument.**

窗体顶端

True

False

窗体底端

 Required

**At which point(s) is a Transaction verifiable?**

窗体顶端

Within it is a TransactionBuilder.

After partial signing.

After complete signing.

All the above.

窗体底端

 Required

**In what form does a TransactionBuilder hold inputs and outputs?**

窗体顶端

TransactionState/StateRef

StateRef/TransactionState

StateRef/StateRef

TransactionState/TransactionState

窗体底端

 Required

**Why does a transaction have to be resolved to a LedgerTransaction before being verified?**

窗体顶端

The inputs must be converted from state references into state objects and the attachments must be converted from hashes into actual attachments

The inputs must be converted from state references into state objects

The inputs must be converted from state references into state objects, the attachments must be converted from hashes into actual attachments, and the TimeWindow must be converted into the node's local time

The transaction's inputs must be checked for double-spends

窗体底端

 Required

**What class does a transaction become after it has been signed?**

窗体顶端

ImmutableTransaction

SignedTransaction

TransactionWithSignatures

WireTransaction

窗体底端

 Required

**What is included in a StateRef to uniquely pinpoint which input state(s) a transaction is spending?**

窗体顶端

The hash of the state

The hash of the transaction that created the state and the hash of the state

The hash of the transaction once everything but the state in question has been filtered out

The hash of the transaction that created the state and the state's index in the outputs of that transaction

窗体底端

 Required

**How is the hash of a transaction calculated?**

窗体顶端

It is the hash of all the transaction's components

It is the hash of all the transaction's components, excluding the TimeWindow

It is the hash of all the transaction's components, excluding the attachments

It is the root hash of the transaction's Merkle tree

窗体底端

 Required

**Why can't a transaction's signatures be checked for validity from within a contract's verify method?**

窗体顶端

We don't know which signature corresponds to which party

The signatures are not included in the transaction being verified

One or more signatures may not yet have been added to the transaction

Because we are verifying a LedgerTransaction instead of a WireTransaction, the hash of the transaction is different, making the signatures invalid

**What role does a command play inside a Transaction?**

窗体顶端

To designate which public keys are required to sign a transaction.

The time period in which a transaction may occur

A hint on the intent of a Transaction and public keys required to sign the transaction.

A hint on the intent of a Transaction.

窗体底端

 Required

**Commands are optional in Corda transactions.**

窗体顶端

True

False

窗体底端

 Required

**How are the required signers of a transaction decided?**

窗体顶端

Based on the participants of the input states

Based on the participants of the input and output states

Based on the signers listed on the commands, plus the participants of the input and output states

Based on the signers listed on the commands

**Which guarantees are provided by Flow framework in Corda?**

窗体顶端

Corda Node won't go offline unless all of the flows are finished

Flows guarantee eventual finality.

Flows are abstractions over messaging operations only. Developers need to checkpoint and restore flows by themselves.

Flows guarantee that every participant can run over a million flows concurrently with less than 1GB RAM consumption

窗体底端

 Required

**How long can flows remain in a suspended state?**

窗体顶端

For a period of time defined by the event horizon specified by the network

While the node is on

Only as long as it is waiting for a message from another node

Indefinitely

**sendAndReceive() method takes a timeout as one of its optional arguments.**

窗体顶端

True

False

窗体底端

 Required

**The Party which initiates a Flow is responsible for creating the output States that are the proposed changes to the ledger.**

窗体顶端

True

False

窗体底端

 Required

**Which of the following about anonymous-to-wellknown party mappings is true?**

窗体顶端

Anonymous-to-wellknown party mappings are resolved as a part of FinalityFlow

Anonymous-to-wellknown party mappings are resolved as a part of CollectSignaturesFlow

Anonymous-to-wellknown party mappings are resolved as a part of IdentitySyncFlow

None of above. Anonymous-to-wellknown party mappings are resolved by Corda core internals

窗体底端

 Required

**Which Corda key concept is the TransactionBuilder used in?**

窗体顶端

State

Contract

Transaction

Flow

窗体底端

 Required

**What pre-defined flow is used to notarise a transaction and record it in every participant/owner's vault?**

窗体顶端

NotarisationFlow

FinalityFlow

LedgerFlow

ServiceFlow

窗体底端

 Required

**What annotation is used to specify which initiator flow a response flow responds to?**

窗体顶端

CounterpartyFlow

ResponderFlow

InitiatedBy

InitiatingFlow

窗体底端

 Required

**What happens if your node receives a message from a flow for which it has not registered a response flow?**

窗体顶端

It runs the node's default flow

It blocks, waiting for the installation of a CorDapp containing a flow that will respond to the initiating flow

It ignores the message

窗体底端

 Required

**When would you wrap an exception in a flow in a FlowException instance?**

窗体顶端

When you want the exception to be propagated back to every node involved in the flow

When you want the exception to be propagated back to every node that is suspended and waiting for your flow to respond because of a receive or sendAndReceive call

When you want the exception to be propagated back to every node involved in the flow except the notary

When you want to hide details of the exception from counterparties for security purposes

窗体底端

 Required

**How is a state's notary changed once the state has been issued?**

窗体顶端

Only the existing notary can change a states's notary using the NotaryChangeFlow

Only a new notary can change a states's notary using the NotaryChangeFlow

The node can change a states's notary using the NotaryChangeFlow

A states's notary cannot be changed

窗体底端

窗体底端

窗体底端

窗体底端

窗体底端

窗体底端

窗体底端

窗体底端

窗体底端