Blockchain CBDE Practice Questions

Shared by Navarro on 21-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

The nonce-field in a transaction is used:

Options:

A- to protect against replay attacks.
B- to have an additional checksum for transactions.
C- to sum up all ethers sent from that address.

Answer:

A

Question 2

Question Type: MultipleChoice

EVM assembly:

Options:

A- is much better than Solidity and a viable alternative.
B- can be split across multiple files, but every contract must be in a file with the same name as the contract itself.
C- is another language similar to LLL, more secure than Solidity.

Answer:

B

Question 3

Question Type: MultipleChoice

All low-level functions on the address, so address.send(), address.call.valueQQ, address.callcode and address.delegatecall:

Options:

A- are interrupting execution on error, because they throw an exception.
B- continuing execution on error silently, which is the reason why they are so dangerous.
C- returning Booleans to indicate an error during execution.
D- .send() throws an exception, while the other functions are returning Booleans during execution to indicate an error.

Answer:

C

Question 4

Question Type: MultipleChoice

Loops in Solidity:

Options:

A- are a great way to circumvent gas requirements, because a loop will only consume gas once.
B- are dangerous when used with data structures that grow, such as arrays or mapping, because it is hard to estimate the gas requirements.
C- should be avoided where possible, because of unknown side-effects on the gas requirements.

Answer:

B

Question 5

Question Type: MultipleChoice

Smart Contracts:

Options:

A- are always living on the same address, because the blockchain is deterministic. So, one account can always have one smart contract.

B- are having the same address as the EOA.
C- are sitting on their own address. The Address is created from the nonce and the EOA address and could be known in advance before deploying the smart contract.
D- the address of the smart contract is a random address which gets generated by the miner who mines the contract-creation transaction.

Answer:

C

Question 6

Question Type: MultipleChoice

To generate a random number:

Options:

A- it's good to use the block timestamp, as this is always different.
B- it's good to use the block hash as this is clearly always very different.
C- it's good to use the RANDAO smart contract.
D- it's not possible to have a random number in a deterministic environment such as the Ethereum blockchain.

Answer:

C

Question 7

Question Type: MultipleChoice

On a consortium network:

Options:

A- everybody can become a miner, everybody can send transactions and everything is public.
B- usually only a few selected nodes can be miners. Transactions can be further limited.

Answer:

B

Question 8

Question Type: MultipleChoice

When a new block is mined:

Options:

A- a list of transactions as well as uncles is incorporated in the block. All gas that is used during those transactions is added to the miners' balance. Also, the block reward is added to the miner. Then the same transactions are run again by every participating node in the network to achieve consensus.
B- a list of transactions is incorporated in that block. Gas used during the execution is attached to the executing contracts while the block reward is automatically spread across the mining pool to ensure a fair spread. Consensus is reached by a special form of hash code.

Answer:

A

Question 9

Question Type: MultipleChoice

Events:

Options:

A- are stored on chain and are a great way to get a return value when a contract calls another contract.
B- are stored in something like a side-chain and cannot be accessed by contracts.
C- are used primarily for debugging exceptions in solidity.

Answer:

B

Question 10

Question Type: MultipleChoice

A Struct is a great way:

Options:

A- to define a new datatype in Solidity, so you don't need to use objects of another contract.
B- to hold instances of other contracts.
C- to implement pointers to other contracts that can hold new datatypes.

Answer:

A

Question 11

Question Type: MultipleChoice

Externally Owned Accounts:

Options:

A- can be destroyed using the selfdestruct keyword. This way all remaining ether will be sent to the receiver address, regardless if they have a fallback function or not.
B- are bound to a private key which is necessary to sign transactions outgoing from that account.
C- are logical opcodes running on the ethereum blockchain very similar to smart contracts.

Answer:

B

Question 12

Question Type: MultipleChoice

Integrating the community into your testing:

Options:

A- is great, because they often find bugs which weren't considered before.
B- is not good, because you might give out secrets.

Answer:

A

Question 13

Question Type: MultipleChoice

Solidity files:

Options:

A- can't be split across multiple files, everything should be in one single file.
B- can be split across multiple files, but every contract must be in a file with the same name as the contract itself.
C- can be spread across multiple files. To import all contract from a file you can use 'import 'myfile.sol'. To import Contract MyContract from myfile.sol you use 'import {MyContract as SomeContract} from 'myfile.sol';'.

Answer:

C