PassQueen
a.k.a dumpsqueen.com

BTA Certified Blockchain Solution Architect

Blockchain CBSA

Version Demo

Total Demo Questions: 28

Total Premium Questions: 282

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

In Hyperledger there are three distinct types of nodes.

What are the three types of nodes? (Select three.)

1. Client Node: That initiates the transaction

2. Peer Nodes: Commits Transaction & keeps the data in sync across the ledger

3. Ordered: They are the communication backbones and responsible for the distribution ofthe transactions.

A. Client Node

B. Orderer Node

C. Certificate Node

D. Anchor

E. Peer Node

ANSWER: A B E

Explanation:

In Hyperledger Fabric, the relevant architectural roles are client applications, peer nodes, and ordering service nodes (orderers). A **Client Node** represents the application-side role that uses an SDK or gateway to submit transaction proposals and, after the required endorsements are obtained, submits transactions for ordering. A **Peer Node** hosts ledgers and smart contracts, endorses proposals when it is configured as an endorsing peer, validates ordered transactions, and commits valid blocks to its ledger. This is how ledger state is independently maintained by the peers in a channel. An **Orderer Node** belongs to the ordering service: it establishes a deterministic transaction order, batches transactions into blocks, and distributes those blocks to channel peers. The ordering service does not execute smart contracts or maintain the peer world state; its core function is reliable ordering and block delivery. Together, these roles separate application interaction, transaction execution and validation, and network-wide transaction ordering—an important design principle in Fabric's permissioned architecture. See the Hyperledger Fabric peer documentation and the Fabric ordering service documentation.

QUESTION NO: 2

What are two specific advantages of using Hyperledger Fabric? (Select two.)

A. No order service needed

B. Use any programming language available

C. Open Source Modular architecture

D. Allows components to be plug-and-play

E. Makes mining cryptos more efficient

ANSWER: C D

Explanation:

Hyperledger Fabric's **open source modular architecture** is a core advantage because Fabric is designed as an enterprise-grade, permissioned distributed-ledger platform whose major services can be configured to meet particular business, privacy, and operational requirements. Its architecture separates concerns such as peer roles, smart-contract execution, ledger management, identity, policy, and transaction ordering, rather than imposing one fixed model for every network.

Fabric also **allows components to be plug-and-play**. The platform supports substitutable implementations of important services, most notably ordering services and membership-service providers. This lets network operators select or integrate components appropriate to their trust model and deployment environment. Fabric's pluggable endorsement and validation policies further allow organizations to define which participants must approve transactions and how transaction rules are enforced. These design choices make Fabric suitable for consortium networks in which participants need tailored governance, confidentiality, and application behavior while sharing a common ledger infrastructure.

The Hyperledger Fabric documentation describes this modularity and its pluggable components in its architecture overview and design documentation: Hyperledger Fabric: What is Fabric? and Hyperledger Fabric Architecture.

QUESTION NO: 3

A chaincode package that was signed at creation can be handed over to other owners for inspection and signing in Hyperledger.

Is it true that the workflow supports out-of-band signing of chaincode package?

A. TRUE

B. FALSE

ANSWER: A

Explanation:

TRUE is correct for the chaincode packaging workflow described in Hyperledger Fabric's legacy chaincode lifecycle documentation. A chaincode deployment package can be created and signed by an authorized organization, then transferred outside the peer-to-peer operational flow to other organizations. Those recipients can inspect the package contents and attach their own signatures before it is installed or deployed. This allows participating organizations to apply their independent governance, review, and approval processes without requiring every signature action to occur through a single online transaction or a shared peer environment.

Out-of-band signing is useful when chaincode owners need to verify the supplied source and metadata, confirm that the package represents the agreed business logic, and provide cryptographic evidence of their approval. The resulting signed package carries the required endorsements from its owners, helping establish confidence that the chaincode being distributed is the artifact that the relevant organizations reviewed. Hyperledger Fabric documents this signed-package workflow and explicitly describes passing a package among owners for inspection and signing. See the Hyperledger Fabric chaincode documentation and the Fabric chaincode lifecycle documentation.

QUESTION NO: 4

Chaincode Services uses Docker to host (deploy) the chaincode without relying on any virtual machine or computer language.

What would be the main reason or best reason that Hyperledger chose containers over virtual machines?

A. Docker provides a secured, lightweight method to sandbox chaincode execution that is not "locked down".

B. Docker provides a secured, lightweight method to sandbox chaincode execution that is not "locked down" but additional programming languages cannot be enabled.

C. Docker provides a secured, lightweight method to sandbox chaincode execution that is "locked down".

D. Docker is fully compatible with Hyperledger and Linux with an upgrade subscription.

ANSWER: C

Explanation:

Docker provides a secured, lightweight method to sandbox chaincode execution that is "locked down". This is the best answer because Hyperledger Fabric's chaincode execution model was designed to isolate smart-contract code from the peer

process and from other deployed chaincodes. A container supplies a bounded runtime environment with the chaincode code, its dependencies, and its language runtime, while avoiding the resource overhead and startup cost of a full virtual machine. That isolation is important because chaincode is application-supplied code: the peer needs to invoke it without giving it unrestricted access to the peer host or the peer's internal process environment.

Fabric documentation describes chaincode as being packaged and deployed for execution by peers, with the chaincode service communicating with the peer through defined interfaces. Container-based execution supports repeatable deployment across organizations: endorsing peers can run the same packaged chaincode and dependency set in a controlled runtime. The "locked down" wording accurately captures the security objective: execution is sandboxed and governed by the platform's configured container environment rather than treated as an unrestricted host-level application. Docker was therefore a practical lightweight isolation mechanism for Fabric's original chaincode runtime architecture. See the Hyperledger Fabric chaincode documentation and the Fabric chaincode launcher documentation.

QUESTION NO: 5

You are currently considering blockchain solutions for your organization. You read an article about "blockless" blockchains.

What are the two benefits that could be gained over a traditional blockchain solution? (Select two.)

A. Faster Block writes

B. Greater Transaction Security

C. Faster Performance

D. Greater transaction Capacity

ANSWER: C D

Explanation:

Faster Performance and Greater transaction Capacity are benefits commonly associated with blockless distributed-ledger designs. Rather than collecting transactions into discrete blocks and requiring the network to wait for the next block-production cycle, these designs can validate, order, or attach transactions more continuously. Eliminating that batching delay can reduce confirmation latency and therefore improve perceived application performance.

Blockless architectures also commonly use directed-acyclic-graph or hashgraph-style data structures and consensus processes that allow many transactions to be handled concurrently. This parallelism can increase the number of transactions the network can process over a given period, producing greater transaction capacity than a conventional chain whose throughput is constrained by block size and block interval. The precise throughput and latency remain dependent on implementation, network conditions, and consensus configuration, but performance and capacity are the intended architectural advantages. IOTA describes its Tangle as a DAG-based distributed ledger and documents its transaction-processing model at IOTA Documentation. Hedera's documentation also explains its hashgraph consensus approach and its high-throughput characteristics at Hedera Hashgraph Consensus.

QUESTION NO: 6

A development team is deploying a Solidity contract that will accept Ether from users.

What are two recommended practices for reducing reentrancy risk? (Select two.)

A. Update internal state before making an external call

B. Use a reentrancy guard where appropriate

C. Use tx.origin for authorization

D. Make all functions public

E. Store private keys in the contract

ANSWER: A B

Explanation:

Update internal state before making an external call and **use a reentrancy guard where appropriate** are correct. Reentrancy can occur when a contract transfers control to an external contract before it has completed its own state updates. The external contract may then call back into the original function and exploit inconsistent state.

The checks-effects-interactions pattern reduces this risk by validating inputs first, updating the contract's internal state second, and interacting with external contracts last. A reentrancy guard can also prevent a protected function from being entered again while it is already executing. These controls should be combined with careful access control, testing, and review of all external calls.

Using *tx.origin* for authorization is not a reentrancy defense and is generally unsafe for access control. See the Solidity security considerations on reentrancy and the OpenZeppelin ReentrancyGuard documentation.

QUESTION NO: 7

Blockchains only work to store financial transactions or other exchanges of monetary value.

A. TRUE

B. FALSE

ANSWER: B

Explanation:

FALSE is correct because a blockchain is a shared, tamper-evident ledger capable of recording many kinds of digitally represented data and state changes, not solely payments or monetary exchanges. Financial transfers were an important early use case, but blockchain solutions can also maintain provenance and chain-of-custody records, verify credentials, coordinate supply-chain events, register assets, manage consent, and execute programmable business logic through smart contracts. The precise data stored varies by architecture: systems may store transaction details directly on-chain, store hashes that prove the integrity of off-chain records, or record references and permissions while sensitive data remains elsewhere. What matters is that network participants can validate the agreed ledger history under the network's consensus and governance rules. This makes blockchain useful where multiple parties need a common, auditable record without relying on each participant maintaining separate, irreconcilable databases. NIST describes blockchain as a tamper-evident and tamper-resistant distributed ledger, a definition that does not limit it to financial data. The Ethereum documentation likewise explains that smart contracts are programs on the blockchain and can support applications beyond currency transfer. See NISTIR 8202: Blockchain Technology Overview and Ethereum smart-contract documentation.

QUESTION NO: 8

If a Proof of Work blockchain such a s Bitcoin or Ethereum changed to a Proof of Stake consensus paradigm, which key component of the Proof of Work process would be eliminated?

A. There would be no need for the miners or nodes to perform a guessing game

B. The need to solve Byzantine Fault Tolerance

C. All fees related to transactions would be removed

D. The blockchain network would no longer have to display public transactions

ANSWER: A

Explanation:

There would be no need for the miners or nodes to perform a guessing game is correct because Proof of Work reaches consensus by having miners repeatedly calculate hashes with different nonce values until one produces a hash satisfying

the network's current difficulty target. This computational race is often described as a guessing game because participants make vast numbers of trial attempts, expending processing power and electricity, to find an acceptable block hash. The valid result demonstrates that work was performed and gives the successful miner the right to propose the next block.

Under Proof of Stake, block-proposal and validation rights are assigned through a staking-based protocol rather than through hash-puzzle competition. Participants lock cryptocurrency as stake, and the protocol selects validators to propose and attest to blocks. Consequently, the energy-intensive nonce-searching/hash-guessing activity that defines Proof of Work is not part of the consensus process. Ethereum's transition to Proof of Stake illustrates this change: validators replaced miners, and block creation no longer depends on mining computations. See the Ethereum Proof-of-Work documentation and the Ethereum Proof-of-Stake documentation.

QUESTION NO: 9

Ethereum is considered to be a _____type of blockchain.

A. Permissionless

B. Permission Based

C. Hybrid

D. Private

ANSWER: A

Explanation:

Ethereum's mainnet is correctly described as **Permissionless**. It is a public blockchain network in which participation is not limited to an approved organization, consortium, or pre-authorized set of users. Anyone with internet access can create an Ethereum account, submit transactions, deploy or interact with smart contracts, inspect the blockchain's public state, and operate infrastructure such as a node. Participation is governed by Ethereum's protocol rules and the payment of required network fees, rather than by an administrator granting membership permission.

Ethereum uses proof-of-stake consensus, under which validators participate by staking ETH and following the protocol's validator requirements. This change in consensus mechanism did not make Ethereum a permissioned network: validator participation remains open to participants who meet the protocol's staking and operational conditions. The network's public, open-access design is fundamental to its role as a shared platform for decentralized applications and smart contracts. Ethereum.org describes Ethereum as a decentralized blockchain with smart-contract functionality and documents how users can run nodes and become validators. See Ethereum.org: What is Ethereum? and Ethereum.org: Nodes and clients.

QUESTION NO: 10

The primary difference between a blockchain and blockless solution is?

A. Blockless solutions are slower

B. Blockless solutions are less reliable

C. Transactions are not verified by the entire network in a blockless environment

D. Blockless solutions do not work on mobile devices

E. Blockless distributed-ledger solutions confirm and record transactions without grouping them into sequential blocks.

ANSWER: E

Explanation:

Blockless distributed-ledger solutions confirm and record transactions without first packaging them into a sequential chain of blocks. This is the defining architectural distinction: a conventional blockchain organizes accepted transactions into blocks, links each block cryptographically to its predecessor, and reaches consensus over the ordered block history. A blockless

design instead uses another data structure and consensus approach, such as a directed acyclic graph or a hashgraph, to establish transaction ordering and finality. This can allow transactions or events to be processed and related more directly, rather than waiting to be included in a miner- or validator-produced block. The specific validation workflow, throughput, and node participation model depend on the particular protocol; they are not what universally defines a system as blockless. Hedera's documentation describes Hashgraph as its consensus algorithm and explains how consensus timestamps and ordering are established for transactions, while Ethereum documentation illustrates the block-based organization of a blockchain ledger. See Hedera Hashgraph consensus documentation and Ethereum's documentation on blocks.

QUESTION NO: 11

A company is designing a blockchain solution that must reference the daily market price of a commodity and automatically settle a contract when the price reaches a defined threshold.

What are two appropriate considerations for using an oracle? (Select two.)

A. An oracle is needed to bring the external market-price data to the smart contract

B. The oracle design is a trust and security dependency

C. Smart contracts can query any public website directly during execution

D. Blockchain consensus guarantees that an oracle's market data is accurate

E. The market price should be hard-coded into the contract for every settlement

ANSWER: A B

Explanation:

An oracle is needed to bring the external market-price data to the smart contract is correct. Smart contracts execute deterministically using data available to the blockchain. They cannot independently make trusted web requests to an external pricing service during execution, because nodes could receive different results and fail to reach consensus. An oracle supplies external data through a transaction or other agreed mechanism.

The oracle design is a trust and security dependency is also correct. If the oracle reports inaccurate, delayed, manipulated, or unavailable data, the smart contract may execute an incorrect outcome even though its on-chain code functions exactly as written. An architecture should assess source integrity, cryptographic authentication, aggregation of independent data sources, monitoring, fallback procedures, and governance for exceptional conditions.

Writing the market price directly into the contract code would not provide current external information, and consensus does not guarantee that an external data source is truthful. Ethereum documentation explains the oracle problem and the need for trusted methods of providing off-chain information: Ethereum.org: Oracles.

QUESTION NO: 12

You are designing a tokenized asset platform that will hold regulated customer information off-chain while recording references and proofs on a consortium blockchain.

What are two sound reasons for storing sensitive personal data off-chain rather than directly on an immutable shared ledger? (Select two.)

A. It reduces exposure of sensitive data to ledger participants

B. It allows correction or deletion under the governance of the off-chain system

C. It makes on-chain records no longer tamper evident

D. It removes the need to secure the off-chain repository

E. It guarantees that hashes and references can never reveal information

ANSWER: A B

Explanation:

It reduces exposure of sensitive data to ledger participants is correct. Even in a permissioned network, replicated ledger data may be retained by multiple organizations and nodes. Storing only a reference, hash, or minimal required data on-chain can limit unnecessary distribution of personal information and support data-minimization requirements.

It allows correction or deletion under the governance of the off-chain system is also correct. An immutable ledger is intentionally difficult to alter after commitment. An off-chain repository can apply retention, correction, access, and deletion policies where legally and operationally required, while the blockchain records a tamper-evident reference or integrity proof. Architects must ensure that the reference itself does not expose sensitive information and must protect the off-chain system with strong access controls and key management.

Encrypting personal data before placing it on-chain does not automatically solve retention, disclosure, key-loss, or future cryptographic-risk concerns. NIST discusses blockchain data-management and privacy considerations in its Blockchain Technology Overview.

QUESTION NO: 13

The most popular Ethereum development framework is currently Truffle.

What are three features of Truffle? (Select three.)

A. Scriptable deployment & migrations framework.

B. Automated contract testing with Mocha and Chai.

C. Takes Dapp transactions via Ws-rpc, json-rpc, ipc-rpc.

D. Built-in smart contract compilation, linking, deployment and binary management.

E. Automated contract testing with Mocha only

ANSWER: A B D

Explanation:

Truffle provides an integrated Ethereum smart-contract development workflow, including compilation, testing, deployment, and artifact handling. “Scriptable deployment & migrations framework.” is a core Truffle capability: migrations are JavaScript-based deployment scripts that let teams deploy contracts in a repeatable, versioned sequence across development, test, and production-style networks. This makes contract rollout more reliable and supports dependencies between deployed contracts.

“Automated contract testing with Mocha and Chai.” is also a documented Truffle feature. Truffle’s test runner supports JavaScript tests written using Mocha, with Chai commonly used for expressive assertions, while connecting the tests to deployed Solidity contracts and a configured blockchain environment.

“Built-in smart contract compilation, linking, deployment and binary management.” correctly describes Truffle’s end-to-end contract tooling. It compiles Solidity sources, manages generated contract artifacts containing ABI and bytecode information, links libraries when necessary, and uses those artifacts during deployment and application interaction. Together, these features provide the development environment, testing framework, and asset pipeline historically described by the Truffle project. See the Truffle GitHub repository and the Truffle documentation archive.

QUESTION NO: 14

A _____cipher basically means it is using a fixed key which replaces the message with a pseudorandom string of characters. It is basically the encryption of each letter one at a time.

What is the cipher type?

A. Stream

B. Block

C. Parallel

D. RSA

ANSWER: A

Explanation:

Stream is the correct cipher type. A stream cipher encrypts plaintext incrementally, typically one bit or byte at a time, by combining each plaintext unit with a corresponding unit from a pseudorandom keystream. In common designs, the keystream is generated from a secret key together with a nonce or initialization value; the resulting keystream is then combined with the message, often using the XOR operation. This matches the question's central description of encrypting the message character by character rather than processing a full fixed-size group of data at once.

Although a secret key is the basis for the encryption, secure real-world use normally requires that a keystream never be reused for different messages under the same key. Reusing a stream-cipher keystream can expose relationships between plaintexts. Modern stream ciphers such as ChaCha20 are therefore used with a unique nonce for each encryption operation. NIST describes stream-cipher encryption as encryption based on a keystream, while the IETF specification for ChaCha20 defines a widely deployed example of this approach. See NIST's stream cipher glossary entry and RFC 8439: ChaCha20 and Poly1305.

QUESTION NO: 15

What are two challenges with using a Proof of Work algorithm? (Select two.)

A. Mining pools not allowed

B. Difficulty rate goes done every year.

C. Expensive

D. Power Intensive

ANSWER: C D

Explanation:

Proof of Work is challenging because it is **Expensive** and **Power Intensive**. In a Proof of Work network, miners repeatedly perform computational hashing in a competition to produce a valid block. This requires specialized hardware, infrastructure, cooling, and ongoing operational maintenance. The capital and operating costs can be substantial, particularly where mining difficulty is high and competition for block rewards is intense.

The same computational competition also consumes significant electricity. Only one participant's valid solution is normally accepted for a block, while the computational work performed by other competing miners does not directly contribute to finalizing that block. Consequently, Proof of Work systems can have substantial energy requirements, and energy availability and price materially affect mining economics. These characteristics are important architectural considerations when selecting a consensus mechanism: Proof of Work provides a well-established mechanism for making attacks costly, but that security model is inherently tied to resource consumption and associated expense. The Bitcoin developer documentation describes mining as the process of expending computational work to create blocks, while the Cambridge Bitcoin Electricity Consumption Index tracks the electricity implications of the Bitcoin Proof of Work network. See Bitcoin Developer Guide: Mining and Cambridge Bitcoin Electricity Consumption Index.

QUESTION NO: 16

Which is the following is the metaphor that describes a logical dilemma that plagues many computer networks?

A. Neo Generals' problem

B. Byzantine Generals' problem

C. Byzantine Admirals' problem

D. Renaissance Generals' problem

ANSWER: B

Explanation:

"Byzantine Generals' problem" is the established metaphor for the challenge of reaching reliable agreement in a distributed system when some participants may fail, send inconsistent information, or act maliciously. The scenario, formalized by Leslie Lamport, Robert Shostak, and Marshall Pease, imagines geographically separated commanders who must coordinate a common decision using messages, while potentially untrustworthy commanders can try to prevent agreement. Its central lesson is that honest participants need a protocol that lets them agree on one outcome despite faulty or adversarial behavior.

This problem is directly relevant to blockchain networks because independent nodes must maintain a consistent shared ledger without relying on a central authority. Byzantine fault-tolerant consensus designs define how nodes validate messages, vote or otherwise coordinate, and continue operating when a bounded portion of participants behaves arbitrarily. Therefore, the metaphor accurately describes the logical dilemma addressed by Byzantine fault tolerance and many blockchain consensus mechanisms. The original paper, "The Byzantine Generals Problem," is available from Leslie Lamport's publication archive. Additional background on Byzantine fault tolerance is provided by IBM Think.

QUESTION NO: 17

In a Bitcoin block, transactions are organized into a structure that allows a node to verify that a transaction is included without downloading every transaction in the block.

What is the name of the hash stored in the block header that represents this transaction structure?

A. Merkle root

B. Transaction nonce

C. Public key hash

D. Witness commitment

E. Coinbase signature

ANSWER: A

Explanation:

Merkle root is correct. Bitcoin transactions in a block are hashed and combined pairwise in a Merkle tree. The final hash at the top of that tree is the Merkle root, and it is included in the block header. Because the header is part of the proof-of-work-protected block, changing a transaction would change its hash, the Merkle root, and ultimately the block header hash.

A node can use a Merkle proof consisting of the relevant neighboring hashes to confirm that a particular transaction contributes to the root. This is useful for simplified payment verification clients because they can verify transaction inclusion using block headers and a compact proof rather than storing complete blocks.

The Bitcoin Developer Guide describes Merkle trees and their use in payment verification: Bitcoin Developer Guide: Simplified Payment Verification.

QUESTION NO: 18

What are some advantages of Proof of Stake(POS) mining over Proof of Work(POW) mining?

(Select three.)

A. Energy efficient in terms of electricity consumption compared to PoW

B. Faster Hashing algorithms

C. No need for expensive hardware compared to PoW

D. Faster validations compared to PoW

E. Better blockchain security compared to POW

ANSWER: A C D

Explanation:

Proof of Stake replaces Proof of Work's competition to solve computational hash puzzles with a process in which validators are selected to propose and attest to blocks based on staked assets and protocol rules. Consequently, it is **energy efficient compared with PoW**, because validators do not need to continuously perform energy-intensive mining computations. Ethereum estimates that its move to Proof of Stake reduced its energy consumption by approximately 99.95%. Proof of Stake also has **no need for expensive hardware compared to PoW**: participating as a validator generally requires standard server or consumer-computer hardware and a reliable network connection rather than specialized ASIC mining equipment. Finally, **faster validations compared to PoW** can be an advantage in Proof of Stake designs because block proposal, validator attestations, and finality can be coordinated without waiting for probabilistic hash-based mining competition. The precise throughput and confirmation time remain protocol parameters, but Proof of Stake supports efficient validation and, in modern implementations, can provide explicit economic finality. These characteristics lower operating costs and hardware barriers while allowing networks to maintain consensus through validator incentives and penalties. See Ethereum's Proof-of-Stake documentation and its overview of Proof-of-Stake energy consumption.

QUESTION NO: 19

A blockchain architect is considering a permissioned consortium network for sharing sensitive trade-finance records among several banks.

What are two benefits of using private data collections in Hyperledger Fabric? (Select two.)

A. Private data can be distributed only to authorized organizations

B. A hash of the private data can be committed to the channel ledger for verification

C. Every peer on the channel receives the private data in clear text

D. Private data collections remove the need for endorsement policies

E. Private data is permanently invisible to every organization, including collection members

ANSWER: A B

Explanation:

Private data can be distributed only to authorized organizations is correct. A private data collection defines a policy identifying which organizations' peers are entitled to receive and retain the actual private data. This supports use cases in which channel members need to share a ledger but not every participant should see every business record.

A hash of the private data can be committed to the channel ledger for verification is also correct. The private-data hash allows authorized organizations and other channel members to validate that the private data used by an authorized peer corresponds to the transaction recorded on the channel ledger, without exposing the private value itself to all channel members.

Private data collections do not eliminate the need for channel membership, identities, endorsement policies, or application-level access controls. They are also distinct from encryption alone: Fabric uses collection policies to control dissemination and retains hashes on the shared ledger. See the Hyperledger Fabric private data documentation.

QUESTION NO: 20

A blockchain architect is reviewing the properties of a digital signature used to authorize transactions.

What are two properties provided by a valid digital signature? (Select two.)

A. Authentication of the signer

B. Integrity of the signed data

C. Encryption of the transaction contents

D. Guaranteed transaction confirmation

E. Recovery of a lost private key

ANSWER: A B

Explanation:

Authentication of the signer and **Integrity of the signed data** are correct. A digital signature is produced using the signer's private key and can be verified using the corresponding public key. Successful verification demonstrates that the signature was created by the holder of the private key, assuming that key has remained under the signer's control.

A signature is also bound to the specific message or transaction data that was signed. If a transaction field is changed after signing, such as its recipient or value, signature verification fails. This makes unauthorized modification evident and is essential in blockchain systems, where nodes independently verify transaction authorization before accepting a transaction.

Digital signatures do not encrypt transaction contents for confidentiality, and they do not by themselves guarantee that an on-chain transaction will be included in a block. See the NIST definition of digital signature and NIST FIPS 186-5 Digital Signature Standard.

QUESTION NO: 21

On the Ethereum blockchain the "nonce" of a transaction:

A. Holds the gas amount to be paid to the miner writing the block that contains the transaction

B. Is used to ensure that transactions by a given account are written in sequential order

C. Is reset to zero after each successfully mined blocked

D. Must always start with four zeroes

ANSWER: B

Explanation:

"Is used to ensure that transactions by a given account are written in sequential order" is correct. In Ethereum, each externally owned account maintains a nonce: a sequential transaction counter. A transaction includes the sender account's expected nonce, and the network accepts it only when it matches the next nonce expected for that sender. Once the transaction is included and processed successfully, the sender's nonce advances by one. This creates an unambiguous ordering for transactions originating from the same account, even though transactions from different accounts do not have a single universal sequence.

The nonce is also a fundamental replay-protection mechanism. Because a specific sender nonce can be used only once in the canonical chain state, a previously executed signed transaction cannot simply be submitted again to produce the same state change. If a sender submits multiple transactions, nonce values establish their required execution order; a later-nonce transaction must wait until preceding nonce values are processed. Ethereum's execution-layer specifications define the transaction nonce as the sender's sequence number, while Ethereum developer documentation describes it as a counter identifying the number of transactions sent from an externally owned account. See the Ethereum Execution Layer Specification and Ethereum.org transaction documentation.

QUESTION NO: 22

Which factor influences the gas cost to deploy a Smart Contact on the Ethereum blockchain?

A. None. Smart Contract deployment has a fixed gas cost

B. The types of operations written in code within the Smart Contract

C. The current Ethereum market conditions

D. The total size of the complied Smart Contract measured in kilobytes

ANSWER: D

Explanation:

The total size of the complied Smart Contract measured in kilobytes influences deployment gas because Ethereum must store the contract's deployed runtime bytecode permanently in the blockchain's state. The Ethereum Virtual Machine charges a code-deposit cost for each byte of runtime code created by a contract-creation transaction. Consequently, a larger compiled contract generally requires more gas to deploy than a smaller one. This is why Solidity developers commonly enable compiler optimization, remove unused functionality, and use libraries or architectural patterns that reduce deployed bytecode when deployment cost matters. Ethereum protocol rules also impose a maximum runtime code size, reinforcing that deployed bytecode size is a material characteristic of contract creation. The stated size is most precisely considered in bytes rather than kilobytes, but the underlying factor is the size of the compiled/deployed contract bytecode. The applicable EVM gas schedule identifies the code-deposit charge as 200 gas per byte, and Ethereum's contract-size-limit proposal describes the runtime-code size constraint. See the EVM CREATE opcode gas documentation and EIP-170: Contract code size limit.

QUESTION NO: 23

In the Ethereum EVM there are two types of memory areas. (Select two.)

A. Storage

B. Database

C. Memory

D. Persistent

E. Ephemeral

ANSWER: A C

Explanation:

Storage and **Memory** are the two relevant EVM data areas described by the question. Storage is the contract's persistent, key-value state: values written there remain available between transactions and calls to the contract. It is part of Ethereum's global state, and modifying it is comparatively expensive because the change must be reflected in the state transition processed by the network.

Memory is temporary, byte-addressable working space created for a single message call. It is used while code executes—for example, to hold intermediate values and dynamically allocated data—and is cleared when that call finishes. Memory can be expanded during execution, with gas charged according to EVM memory-expansion rules, but it does not preserve data for later transactions.

Solidity's documentation distinguishes these locations because variable location affects both lifetime and gas behavior: persistent contract state belongs in storage, while temporary execution data belongs in memory. The EVM specification also defines separate operational semantics and gas costs for memory and persistent storage operations. See the Solidity documentation on storage, memory, and the stack and the Ethereum EVM overview.

QUESTION NO: 24

What are two reasons that you would consider implementing a POW algo in your blockchain? (Select two.)

A. PoW imposes no limits on actions in the network and therefore can thwart attacks better than other algos due to high cost

B. What matters is to have large computational power to solve the puzzles and form new blocks over having a financial stake.

C. PoW imposes some limits on actions in the network and therefore can thwart attacks better than other algos due to high cost

D. The algo is energy efficient compared to POS and BFT

E. The algo is energy efficient compared to POS and DPOS

ANSWER: B C

Explanation:

PoW imposes some limits on actions in the network and therefore can thwart attacks better than other algos due to high cost is correct because proof of work requires a participant to expend computational effort before producing a valid block. This makes rewriting history, producing competing chains, or sustaining certain forms of network abuse economically expensive: an attacker needs substantial hashing capacity and must continually pay for the associated electricity and hardware operation. Bitcoin's developer documentation describes proof of work as a mechanism that makes block creation costly and ties chain security to accumulated work.

What matters is to have large computational power to solve the puzzles and form new blocks over having a financial stake is also correct. In a PoW consensus system, block-production probability is determined by a miner's share of the network hash rate rather than by the number of native tokens held. This can be a relevant design choice where the network wants consensus participation and security weight to be based on externally verifiable computational work instead of token ownership. The mining process repeatedly hashes candidate block headers until a hash satisfies the current target, with successful miners earning the opportunity to propose blocks. See the Bitcoin Developer Guide: Block Chain and the Bitcoin Developer Reference: Proof of Work.

QUESTION NO: 25

What benefit does HPE Cloud Cruiser provide?

A. Need for proven, production-ready technology

B. Need a full history of all application data

C. Require top-notch performance

D. Need to avoid data duplication or redundancy

E. Cloud-consumption cost visibility, metering, and chargeback reporting

ANSWER: E

Explanation:

HPE Cloud Cruiser provides cloud-consumption management: it collects and normalizes usage data across cloud environments, associates that consumption with costs, and supports financial reporting such as showback and chargeback. This gives IT and business teams visibility into who is using cloud services, what those services cost, and how spending can be governed or optimized. Its value is therefore financial transparency and accountability for hybrid-cloud consumption, rather than a blockchain data-history, performance, redundancy, or production-readiness capability. HPE described its acquisition of Cloud Cruiser as an extension of hybrid-cloud management that adds cloud analytics, metering, and billing capabilities. These functions are essential when an organization needs to allocate costs to departments, customers, applications, or projects based on actual consumption. See HPE's announcement, HP acquires Cloud Cruiser, and the HPE cloud management overview.

QUESTION NO: 26

In regard to blockchain, complete autonomy, transparency, and decentralization:

A. Are the aim of all distributed banking and financial institutions

B. Will never be achievable due to technology limitations

C. Are highly valuable in some use cases and completely undesirable in other use cases

D. Are the foundation principles that allow blockchain to operate successfully

ANSWER: D

Explanation:

“Are the foundation principles that allow blockchain to operate successfully” is the correct choice because blockchain systems are designed to coordinate records and transactions among multiple participants without requiring a single central operator to maintain the authoritative ledger. Decentralization distributes validation and recordkeeping across participating nodes, helping create resilience and reducing reliance on one intermediary. Transparency means that authorized network participants can inspect and independently verify the shared ledger’s transaction history and current state; the precise degree of visibility may be adjusted in permissioned implementations. Autonomy is supported where protocol rules and smart contracts execute agreed logic deterministically once their conditions are met, rather than depending on a manual intermediary for every action. Together, these characteristics enable participants who may not fully trust one another to use shared, verifiable data and consistent rules. Ethereum’s documentation describes blockchain as a shared, decentralized database and explains how smart contracts run according to code deployed on the network. The NIST overview likewise identifies decentralization and distributed consensus as central blockchain concepts. See Ethereum developer documentation and NISTIR 8202: Blockchain Technology Overview.

QUESTION NO: 27 - (SIMULATION)

When developing in Ethereum which is considered to be an In-Memory Blockchain simulations for rapid development?

ANSWER: Check the explanation for the answer

Explanation:

Correct answer: `TestRPC` (now commonly called `Ganache`).

TestRPC/Ganache is an in-memory Ethereum blockchain simulator used for rapid local development and testing. It simulates accounts, transactions, blocks, and mining without requiring connection to a live Ethereum network.

The listed in-memory simulation options are:

```
TestRPC
Ganache
Truffle Developer Console
```

QUESTION NO: 28

It is impossible to persist media types such as MP3 and MOV on any distributed ledger.

A. FALSE

B. TRUE

ANSWER: A

Explanation:

FALSE is correct because distributed-ledger systems can persist arbitrary digital data, including the byte content of media files. A ledger does not inherently prohibit a particular file format; MP3 and MOV files are ultimately sequences of bytes, which can be encoded and written into transactions, smart-contract storage, or other ledger-supported data fields where the platform permits it.

In practice, storing complete media files directly on a blockchain is usually a poor architectural choice. Replicating large files across many nodes can be expensive, consume block capacity, and reduce network efficiency. Consequently, blockchain solutions commonly store a cryptographic hash, content identifier, ownership record, access policy, or timestamp on the ledger, while storing the larger media object in decentralized storage or another external repository. The on-ledger hash provides an integrity commitment: a recipient can hash the retrieved file and verify that it matches the value recorded on the ledger.

This architectural preference is about cost, scalability, and operational design—not impossibility. Ethereum's documentation describes persistent smart-contract storage, while IPFS documents content addressing, a common complement for retaining and verifying large digital assets: Solidity documentation: storage, memory, and the stack and IPFS documentation: content addressing.