ETE

Here are common types of questions for each topic:

BT-7

1. **Cryptographic Hash Functions**
    - Define a cryptographic hash function.
    - What is collision resistance? Why is it important?
    - Example: Explain how SHA-256 is used in blockchain.
2. **Merkle Trees**
    - What is the purpose of a Merkle tree in blockchain?
    - How can you prove a specific transaction exists in a Merkle tree?
    - Example: Illustrate the structure of a Merkle tree for four transactions.
3. **Proof of Work**
    - What is the proof-of-work mechanism?
    - How does adjusting difficulty (D) influence mining?
    - Example: Describe the steps to verify a proof-of-work solution.
4. **Digital Signatures**
    - Explain the role of public and private keys in digital signatures.
    - What are the advantages of ECDSA over RSA?
    - Example: How does a digital signature ensure transaction authenticity?

1. Theory-Based Questions

- Define cryptographic hash functions and give an example.
- Explain the properties of collision-resistant hash functions.
- What is a Merkle tree? Why is it used in blockchain?

2. Application-Based Questions

- How does a Merkle tree help prove the validity of transactions?
- Describe the proof-of-work process used in Bitcoin.
- Explain how digital signatures prevent tampering in blockchain transactions.

---

3. Numerical/Diagrammatic Questions

- Construct a Merkle tree for given transactions.
- If a hash function outputs 256-bit values, what is the probability of collision after 2^{64} inputs?

- Given a hash puzzle H(x,y)<2256/DH(x, y) < 2^{256}/DH(x,y)<2256/D, calculate yyy for a given xxx and DDD.

---

4. Comparison/Analysis Questions

- Compare symmetric and asymmetric encryption.
- What are the benefits of ECDSA over RSA in blockchain?
- How do BLS signatures improve scalability in Ethereum 2.0?

---

5. Problem-Solving Questions

- Prove that a transaction belongs to a blockchain using Merkle proofs.
- Design a digital signature scheme for secure message verification.
- Analyze the efficiency of proof-of-work under increasing difficulty.

BT-8

1. Theory-Based Questions

- Explain the key differences between Bitcoin and Ethereum.
- What is a Merkle Patricia Trie? How is it used in Ethereum?
- Define smart contracts and explain their lifecycle in Ethereum.
- What are the types of Ethereum accounts?

2. Application-Based Questions

- How does a Merkle tree help prove the validity of transactions?
- Describe the proof-of-work process used in Bitcoin.
- Explain how digital signatures prevent tampering in blockchain transactions.

---

3. Numerical/Diagrammatic Questions

- Construct a Merkle tree for given transactions.

- If a hash function outputs 256-bit values, what is the probability of collision after 2^{64} inputs?
- Given a hash puzzle H(x,y)<2256/DH(x, y) < 2^{256}/DH(x,y)<2256/D, calculate yyy for a given xxx and DDD.

---

4. Comparison/Analysis Questions

- Compare symmetric and asymmetric encryption.
- What are the benefits of ECDSA over RSA in blockchain?
- How do BLS signatures improve scalability in Ethereum 2.0?

---

5. Problem-Solving Questions

- Prove that a transaction belongs to a blockchain using Merkle proofs.
- Design a digital signature scheme for secure message verification.
- Analyze the efficiency of proof-of-work under increasing difficulty.

1. Conceptual Questions

- What are the limitations of Bitcoin that Ethereum overcomes?
- Explain the role of the Ethereum Virtual Machine (EVM) in executing smart contracts.
- Describe the difference between externally owned accounts (EOAs) and contract accounts (CAs).

---

2. Practical Understanding

- Why does Ethereum use gas for transactions? How does it affect network security?
- Explain the significance of nonce in Ethereum transactions.
- How does the “base fee” in Ethereum prevent transaction spamming?

---

3. Advanced Topics

- What is the role of uncle blocks in Ethereum, and how are they rewarded?
- Discuss the use of ERC standards (ERC-20, ERC-721) in Ethereum applications.
- Explain the concept of composability in Ethereum smart contracts with examples.

---

4. Solidity and Smart Contracts

- Write a Solidity function to transfer tokens between two addresses.
- What are the advantages of using Solidity for smart contract development?
- Explain how NameCoin demonstrates the use of smart contracts for domain registration.

---

5. Blockchain Architecture

- How are transactions, states, and receipts stored in Ethereum?
- Explain the components of an Ethereum block body and their importance.
- How does Ethereum handle state transitions differently from Bitcoin?

---

6. Real-Life Applications

- Discuss how Ethereum enables DeFi applications like lending and stablecoins.
- What is the role of Merkle Patricia Trie in Ethereum storage and verification?
- How does Ethereum facilitate decentralized apps (DApps) through smart contracts

---

BT-8.1

Theory-based Questions:

- Definitions, concepts, and principles (e.g., "Define Ethereum Virtual Machine").
- Comparisons (e.g., "Compare blockchain and traditional databases").

Application-based Questions:

- Use case scenarios (e.g., "Explain the role of a miner in the Ethereum network").
- Implementation details (e.g., "How is a transaction executed in Ethereum?").

Problem-solving Questions:

- Calculations or design (e.g., "Design a smart contract for a voting system").
- Debugging code or analyzing blockchain transactions.

Diagram-based Questions:

- Explaining systems with diagrams (e.g., "Draw and explain the architecture of Ethereum").

Short Notes or Essays:

- Writing detailed explanations about specific topics (e.g., "Write a short note on GAS in Ethereum").

1. Objective-type Questions

- **MCQs**: Multiple choice questions for quick assessments (e.g., "Which consensus algorithm does Ethereum use by default?").
- **True/False**: For simple conceptual checks (e.g., "Blockchain is a centralized system: True or False").
- **Fill-in-the-blanks**: Testing key terms (e.g., "The smallest unit of Ether is called _____.").

---

2. Practical-based Questions

- **Code Analysis**: Examine provided code snippets and predict outputs or identify errors (e.g., "What will the output of this Solidity function be?").
- **Scenario-based Tasks**: Create or modify a system (e.g., "Write a smart contract for a crowdfunding platform").

---

3. Case Studies or Real-world Scenarios

- Students analyze a real-world application or problem and explain solutions (e.g., "Discuss how Ethereum can be used to improve supply chain management").
- Evaluate a situation and provide insights (e.g., "Analyze the pros and cons of Ethereum's switch to Proof of Stake in the context of energy efficiency").

---

4. Research-oriented Questions

- Questions encouraging exploration of recent advancements (e.g., "Explain how Layer 2 scaling solutions, such as Optimistic Rollups, enhance Ethereum's performance").
- Summary of research papers (e.g., "Discuss the core points of the Ethereum Yellow Paper").

---

5. Critical Thinking and Open-ended Questions

- Encourage debate or opinion-based answers (e.g., "Will blockchain disrupt traditional banking systems? Justify your answer").

- Hypothetical scenarios (e.g., "If Ethereum were to replace fiat currency, what changes would you expect in the global economy?").

---

6. Diagrammatic and Tabular Questions

- Require students to draw diagrams or flowcharts (e.g., "Illustrate the architecture of the Ethereum Virtual Machine (EVM)").
- Tabular comparisons (e.g., "Compare Ethereum and Bitcoin in terms of purpose, consensus, and scalability").

---

7. Interdisciplinary Questions

- Combining knowledge from different fields (e.g., "Discuss how blockchain principles can be applied in healthcare for maintaining patient data")

1. Multiple Choice Questions (MCQs)

1. **What type of state machine does Ethereum use?**
   - a) Rule-based
   - b) Transaction-based
   - c) Sequential
   - d) Temporal

   Answer: b) Transaction-based
2. **Which account type in Ethereum is controlled by a private key?**
   - a) Contract Account
   - b) Externally Owned Account (EOA)
   - c) Miner Account
   - d) System Account

   Answer: b) Externally Owned Account (EOA)
3. **What is the primary purpose of the GAS in Ethereum?**
   - a) Measure computation resources
   - b) Store transactions
   - c) Encrypt private keys
   - d) Validate accounts

   Answer: a) Measure computation resources

4. **Which consensus algorithm was Ethereum originally based on?**

- a) Proof of Stake
- b) Proof of Work
- c) Delegated Proof of Stake
- d) Byzantine Fault Tolerance
  Answer: b) Proof of Work

---

2. True/False Questions

1. Ethereum uses a decentralized peer-to-peer (P2P) network.
   Answer: True
2. Contract accounts in Ethereum are controlled by private keys.
   Answer: False
3. The Ethereum Virtual Machine (EVM) is stack-based.
   Answer: True
4. The nonce in a transaction is used to prevent double-spending.
   Answer: True

---

3. Fill-in-the-Blanks

1. Ethereum can be viewed as a _____-based state machine.
   Answer: transaction
2. The smallest unit of Ether is called _____.
   Answer: Wei
3. A blockchain can be described as a _____, decentralized database.
   Answer: globally shared
4. The two types of accounts in Ethereum are _____ and _____.
   Answer: Externally Owned Account (EOA), Contract Account

---

4. Match the Following

Column A	Column B
GAS	Measures computational costs

Externally Owned Account	Controlled by private key
Contract Account	Controlled by EVM code
SHA-3	Ethereum's hashing algorithm

---

5. Assertion and Reasoning

1. **Assertion (A)**: Ethereum is a blockchain platform that supports smart contracts.
   Reason (R): Smart contracts in Ethereum are written using the Solidity programming language.
   - a) Both A and R are true, and R is the correct explanation of A.
   - b) Both A and R are true, but R is not the correct explanation of A.
   - c) A is true, but R is false.
   - d) A is false, but R is true.
     Answer: a) Both A and R are true, and R is the correct explanation of A.

BT-8

Types of Questions

1. Objective Questions

1. **Multiple Choice Questions (MCQs)**:
   - What is the output size of SHA-256?
     a) 64 bytes
     b) 32 bytes
     c) 16 bytes
     d) 128 bytes
     Answer: b) 32 bytes
   - Which property ensures that it is "hard" to find a collision in a hash function?
     a) Compression
     b) Determinism
     c) Collision resistance

d) Preimage resistance
Answer: c) Collision resistance

2. **True/False**:
   - Digital signatures ensure both integrity and authenticity.
     Answer: True
   - RSA is the most commonly used signature algorithm in blockchains like Bitcoin and Ethereum.
     Answer: False (ECDSA is used).
3. **Fill-in-the-Blanks**:
   - The _____ algorithm is used for digital signatures in Bitcoin.
     Answer: Schnorr
   - Ethereum 2.0 employs _____ signatures for efficient validation.
     Answer: BLS (Boneh-Lynn-Shacham)
4. **Match the Following**:

Concept	Description
Merkle Tree	Data structure for verifying transactions
Collision Resistance	Property of cryptographic hash functions
Digital Signature	Ensures integrity and authenticity
Proof of Work	Consensus mechanism

---

2. Analytical or Problem-solving Questions

1. Explain how Merkle trees ensure data integrity in a blockchain.
2. Given a cryptographic hash function HHH, prove why collision resistance is essential for blockchain applications.

---

3. Diagram-based Questions

1. Draw and explain the structure of a Merkle tree for four transactions T1,T2,T3,T4T1, T2, T3, T4T1,T2,T3,T4.
2. Illustrate the process of generating and verifying a digital signature using public and private keys.

---

4. Scenario-based or Application Questions

1. Describe how digital signatures are used to validate transactions in blockchain systems.
2. In a Proof-of-Work system, explain why increasing the difficulty parameter DDD makes mining computationally expensive.

---

5. Research-oriented or Case Study Questions

1. Compare ECDSA, Schnorr, and BLS signature schemes in terms of scalability and efficiency for blockchain use cases.
2. Discuss the impact of quantum computing on cryptographic techniques like RSA and ECDSA.

Objectives for Questions

1. **Conceptual Understanding**:
    - Test foundational knowledge of cryptography (e.g., properties of hash functions, digital signatures).
2. **Application of Knowledge**:
    - Assess the ability to apply cryptographic concepts to real-world blockchain scenarios (e.g., use of Merkle trees in transaction verification).
3. **Critical Thinking**:
    - Evaluate analytical reasoning and problem-solving skills (e.g., impact of collision resistance failure on blockchain security).
4. **Technical Proficiency**:
    - Test practical knowledge, such as signature verification or building Merkle proofs.
5. **Research and Comparison**:
    - Explore advancements like post-quantum cryptography and how blockchain protocols adapt to emerging threats

Multiple Choice Questions (MCQs)

1. What is the primary advantage of a Merkle tree in blockchain systems?
    - a) Reduces computational complexity
    - b) Ensures faster block mining

    - c) Efficient verification of data integrity
    - d) Eliminates the need for digital signatures
      Answer: c) Efficient verification of data integrity
2. Which property of digital signatures ensures that a signed message cannot be forged?
    - a) Non-repudiation
    - b) Confidentiality
    - c) Scalability
    - d) Data compression
      Answer: a) Non-repudiation
3. Proof of Work (PoW) is best described as:
    - a) A cryptographic hash puzzle that is difficult to solve but easy to verify
    - b) A method to encrypt transactions in a blockchain
    - c) A consensus algorithm that uses voting
    - d) A way to store blocks efficiently
      Answer: a) A cryptographic hash puzzle that is difficult to solve but easy to verify
4. In Ethereum, which cryptographic algorithm is used for hashing?
    - a) SHA-1
    - b) SHA-256
    - c) Keccak-256
    - d) Blake2
      Answer: c) Keccak-256

True/False Questions

1. A Merkle root is the hash of all transactions in a block, organized in a binary tree.
   Answer: True
2. ECDSA is no longer used in Bitcoin after the switch to Proof-of-Stake.
   Answer: False
3. In Proof of Work, the difficulty parameter DDD controls how long it takes to mine a block.
   Answer: True
4. Public key cryptography requires the same key for encryption and decryption.
   Answer: False

Fill-in-the-Blanks

1. A _____ is a cryptographic function that maps input data of arbitrary size to a fixed size.
   Answer: hash function
2. The _____ algorithm ensures that data integrity is maintained in digital communications.
   Answer: digital signature

3. The _____ structure in blockchain allows verification of any transaction in logarithmic time.
   Answer: Merkle tree
4. _____ cryptography uses two keys: a public key and a private key.
   Answer: Asymmetric

Analytical Questions

1. **Explain the role of collision resistance in cryptographic hash functions. Provide an example of its application in blockchain systems.**
2. **Compare Proof of Work (PoW) and Proof of Stake (PoS) consensus mechanisms. Highlight their strengths and weaknesses.**

Scenario-based Questions

1. **Describe how a digital signature can be used to secure a financial transaction on the blockchain. Include steps for signature generation and verification.**
2. **You are given a Merkle root of a block. Explain how you would verify that a specific transaction is part of this block.**

Diagram-based Questions

1. **Illustrate the structure of a Merkle tree with five transactions (T1,T2,T3,T4,T5T1, T2, T3, T4, T5T1,T2,T3,T4,T5) and show the process to verify T3T3T3.**
2. **Draw and explain the process of a digital signature from key generation to signature verification.**

Case Study Questions

1. **How does Ethereum 2.0 use BLS signatures to improve scalability? Explain the advantages over traditional ECDSA.**
2. **Discuss the use of Proof of Work (PoW) in Bitcoin. Why is it energy-intensive, and what are the potential alternatives?**

Open-ended Questions

1. **What are the potential security risks of using older cryptographic algorithms, such as SHA-1, in modern blockchain systems?**
2. **In a post-quantum computing era, how should blockchain systems evolve to maintain security and integrity?**

BT-5

Objective Questions

Multiple Choice Questions (MCQs)

1. **What is the main principle of the Nakamoto Consensus?**
    - a) Proof of Stake and sharding
    - b) Leader-based block approval
    - c) Proof of Work and the longest chain rule
    - d) Byzantine Fault Tolerance

    Answer: c) Proof of Work and the longest chain rule
2. **Which factor determines the mining difficulty in Proof of Work?**
    - a) Number of transactions in a block
    - b) Number of leading zeros in the target hash
    - c) Hashing algorithm used
    - d) Blockchain size

    Answer: b) Number of leading zeros in the target hash
3. **What does the Nakamoto property of "chain quality" ensure?**
    - a) Consistency among all miners
    - b) Percentage of honestly mined blocks
    - c) Reduction in mining costs
    - d) Guaranteed block finality

    Answer: b) Percentage of honestly mined blocks
4. **What is the primary drawback of Proof of Work?**
    - a) Centralized decision-making
    - b) High energy consumption
    - c) Lack of security
    - d) Inability to prevent DoS attacks

    Answer: b) High energy consumption
5. **What is a "51% attack"?**
    - a) When a blockchain reaches 51 blocks
    - b) When an attacker controls more than half of the network's computational power
    - c) When a blockchain splits into multiple forks
    - d) When miners collaborate to reduce rewards

    Answer: b) When an attacker controls more than half of the network's computational power

---

True/False

1. The Nakamoto Consensus can tolerate up to 50% corruptions in the network.
   Answer: False (It tolerates up to less than 50%).
2. Proof of Work uses computational puzzles to ensure the integrity of blockchain transactions.
   Answer: True
3. Selfish mining requires an attacker to control more than 50% of the mining power.
   Answer: False (It can work with as little as 33%).
4. Orphan blocks are always included in the final blockchain.
   Answer: False

Fill-in-the-Blanks

1. Nakamoto Consensus combines _____ and the _____ rule to secure blockchain networks.
   Answer: Proof of Work, longest chain
2. In Proof of Work, the computational puzzle difficulty is adjusted based on _____.
   Answer: network load and mining rate
3. The _____ property ensures that honest miners contribute the majority of blocks in a blockchain.
   Answer: chain quality
4. A _____ attack occurs when an attacker keeps a private fork to invalidate honest miners' blocks.
   Answer: selfish mining

Analytical or Application Questions

1. **Explain the relationship between mining difficulty and blockchain security in Proof of Work systems.**
2. **Analyze how a 51% attack can affect the blockchain network. Provide examples from real-world incidents.**
3. **Why does Nakamoto Consensus require blocks to have a delay between proposals? How does it maintain network stability?**

Diagram-based Questions

1. **Draw and explain the Proof of Work mining process, including the role of the nonce and the target hash.**

2. **Illustrate how a selfish miner operates to gain an advantage over honest miners in a blockchain network.**

Scenario-based Questions

1. **A blockchain network is experiencing frequent forks. How can Nakamoto Consensus properties, like chain quality and consistency, help reduce such issues?**
2. **In a Proof of Work system, explain the impact of adjusting the block time from 10 minutes to 5 minutes on network security and efficiency.**

Research and Case Study Questions

1. **Discuss the energy consumption issues associated with Proof of Work and compare it with Proof of Stake as an alternative.**
2. **Evaluate the Nakamoto Consensus in handling network delays. What are its limitations, and how can they be addressed?**

Open-ended Questions

1. **Can Nakamoto Consensus be scaled for high-throughput applications like DeFi? Discuss its challenges and potential solutions.**
2. **What would happen if miners collaborated to form a centralized mining pool controlling 60% of the network? Analyze its implications on decentralization**

Objective Questions

Multiple Choice Questions (MCQs)

1. **What is the key property of Nakamoto Consensus that ensures blocks are added to the longest chain?**
    - a) Fork resolution
    - b) Leader election
    - c) Mining rewards
    - d) Proof of Work

      Answer: d) Proof of Work
2. **What happens to orphaned blocks in a blockchain network?**
    - a) They are included in the chain later.
    - b) They remain unused but visible.

    - c) They are permanently discarded.
    - d) They are merged into the longest chain.
      Answer: b) They remain unused but visible.
3. **Which of the following is an example of a "double-spending attack"?**
    - a) Deliberate forking to censor transactions
    - b) Spending the same cryptocurrency on two different transactions
    - c) Selfish mining to invalidate honest blocks
    - d) Changing the hash function used by the network
      Answer: b) Spending the same cryptocurrency on two different transactions
4. **What does "chain growth" in Nakamoto Consensus signify?**
    - a) The blockchain can accommodate more nodes.
    - b) Honest blocks are added at a steady rate.
    - c) Network delays are minimized.
    - d) Forks are automatically resolved.
      Answer: b) Honest blocks are added at a steady rate.
5. **What determines the difficulty of mining in a Proof of Work system?**
    - a) Size of transactions in the block
    - b) Total number of miners in the network
    - c) Target hash and network load
    - d) Blockchain size
      Answer: c) Target hash and network load

True/False Questions

1. The Nakamoto Consensus guarantees finality after kkk blocks.
   Answer: True
2. A selfish mining attack requires the attacker to control at least 51% of the network's computational power.
   Answer: False (It can work with as little as 33%).
3. Mining pools reduce the variance in mining rewards for individual miners.
   Answer: True
4. Consistency in Nakamoto Consensus means all nodes must have identical copies of the blockchain at all times.
   Answer: False (Consistency tolerates minor delays for kkk blocks).

Fill-in-the-Blanks

1. _____ ensures that only one valid block is added to the blockchain at a time.
   Answer: Proof of Work

2. A _____ is a block mined at the same time as another block but not included in the longest chain.
   Answer: fork (or orphan block)
3. _____ mining is an attack strategy where a miner keeps a private chain to maximize rewards.
   Answer: Selfish
4. The Nakamoto Consensus is designed to tolerate up to _____ corruptions in the network.
   Answer: less than 50%

Analytical Questions

1. **Explain why a "51% attack" is considered the most severe threat to Proof of Work blockchains. Discuss the implications for decentralization.**
2. **Analyze the relationship between mining difficulty and block time in a Proof of Work blockchain. How does adjusting the difficulty impact network performance and security?**

Diagram-based Questions

1. **Illustrate the structure of a Proof of Work block, including components like nonce, timestamp, and Merkle root. Explain how the hash is computed.**
2. **Draw and explain the effects of a selfish mining attack on the blockchain. Highlight how the attacker's chain can overtake the honest chain.**

Scenario-based Questions

1. **A blockchain network with a 10-minute block time is experiencing high network delays. Discuss how this affects the Nakamoto Consensus properties, including consistency and chain growth. Suggest mitigation strategies.**
2. **Suppose an adversary controls 40% of the network's computational power. How likely is it for the adversary to successfully execute a double-spend attack? Use Nakamoto Consensus properties to justify your answer.**

Research and Case Study Questions

1. **Evaluate the energy consumption issues associated with Proof of Work. Compare it to Proof of Stake and analyze whether PoS can fully address PoW's inefficiencies.**
2. **Study the impact of mining pools on the decentralization of Bitcoin. How do they influence the Nakamoto Consensus's robustness against attacks?**

Open-ended Questions

1. **Can Nakamoto Consensus adapt to faster block times without sacrificing security? Discuss the trade-offs involved.**
2. **If the majority of miners shifted to renewable energy sources, would Proof of Work become a sustainable consensus mechanism? Why or why not**