Blockchain Basics Notes

Here are detailed notes for **Module 1: Basics of Blockchain** to help you prepare for your 60-mark exam.

Module 1: Basics of Blockchain

1. Distributed Database

- A distributed database stores data across multiple systems or locations.
- Ensures fault tolerance and availability.
- Blockchain is a type of distributed database where all nodes share the same data.

2. Two General Problems

- **Double Spending**: Using the same digital token multiple times.
- **Consensus Problem**: Agreement among distributed systems on a single data value.

3. Byzantine General Problem

- A scenario where parties must agree on a strategy, but some may be unreliable.
- Solved using consensus algorithms (e.g., PoW, PoS).

4. Fault Tolerance

- The system continues to operate despite failures.
- Blockchain achieves this via redundancy and consensus.

5. Advantages over Conventional Databases

- Decentralized, tamper-proof, immutable, transparent.
- No single point of failure.

6. Blockchain Network

- A peer-to-peer network of nodes.
- Each node maintains a full copy of the blockchain.

7. Mining Mechanism

- Process of adding blocks to the blockchain.

- Involves solving complex mathematical problems (PoW).
- Miners receive rewards for their work.

8. Distributed Consensus

- Ensures all copies of the ledger are synchronized.
- Methods include Proof of Work (PoW), Proof of Stake (PoS), etc.

9. Merkle Patricia Tree

- A hash-based data structure used in Ethereum.
- Allows quick and secure verification of transactions.

10. Gas Limit, Transactions and Fees

- **Gas Limit**: Max computational steps for a transaction.
- **Fees**: Incentivize miners and prevent spam.

11. Anonymity & Reward

- Blockchain supports pseudonymity (not full anonymity).
- Miners or validators are rewarded with cryptocurrency.

12. Chain Policy

- Rules for block creation, validation, and rewards.

13. Life of Blockchain Application

- Starts with concept -> Development -> Mining -> Usage -> Forks -> Upgrades.

14. Soft Fork vs Hard Fork

- **Soft Fork**: Backward-compatible protocol update.
- **Hard Fork**: Non-backward-compatible, splits the chain into two.

15. Private and Public Blockchain

- **Public**: Open to everyone (e.g., Bitcoin, Ethereum).
- **Private**: Restricted access, often used in enterprises.

Let me know if you want me to continue with **Module 2: Distributed Ledger Technology**.

Here are detailed notes for **Module 2: Distributed Ledger Technology**:

Module 2: Distributed Ledger Technology

1. **Cryptography**

- Used to secure transactions and ensure data integrity.
- Core techniques: hashing, digital signatures, and encryption.

2. **Hash Function**

- Converts input into a fixed-length hash.
- Properties: deterministic, fast computation, collision-resistant, pre-image resistant.
- Example: SHA-256 used in Bitcoin.

3. **Digital Signature – ECDSA**

- Ensures authenticity and non-repudiation of messages.
- **ECDSA** (Elliptic Curve Digital Signature Algorithm) is efficient and secure.
- Used to sign transactions in Bitcoin and Ethereum.

4. **Memory Hard Algorithm**

- Requires significant memory to compute.
- Resists ASIC-based mining (e.g., Scrypt used in Litecoin).

5. **Zero Knowledge Proof**

- Proves possession of knowledge without revealing the information.
- Useful in privacy-focused blockchains (e.g., Zcash).

6. **Digital Trust, Asset, and Transactions**

- **Digital Trust**: Trust established through cryptographic verification, not intermediaries.
- **Asset**: Anything of value (currency, documents, etc.).
- **Transaction**: A record of asset transfer on the ledger.

7. **Distributed Ledger Technology (DLT)**

- A decentralized database maintained by multiple participants.

- Blockchain is one type of DLT.
- Eliminates intermediaries and enhances transparency.

8. Types of Network Components

- **Nodes**: Participants maintaining the ledger.
- **Validators**: Nodes validating transactions.
- **Miners**: Solve puzzles to create blocks.
- **Clients**: Initiate transactions.

9. Blockchain vs DLT

- All blockchains are DLTs, but not all DLTs are blockchains.
- **Blockchain**: Adds data in chained blocks.
- **DLT**: Broader; can store data in various formats.

10. Ledger Types

- **Permissionless (Public)**: Open to all, e.g., Bitcoin.
- **Permissioned (Private)**: Controlled access, e.g., Hyperledger.
- **Consortium**: Controlled by a group, hybrid type.

Let me know when you're ready for **Module 3: Distributed Consensus**.

Here are detailed notes for **Module 3: Distributed Consensus**:

Module 3: Distributed Consensus

1. Nakamoto Consensus

- Used in Bitcoin.
- Combines Proof of Work (PoW) with longest chain rule.
- Forks are resolved by choosing the longest valid chain.

2. Proof of Work (PoW)

- Miners solve cryptographic puzzles.
- First to solve adds the next block.
- Energy-intensive but secure against Sybil attacks.

3. **Proof of Stake (PoS)**
- Validators are chosen based on their stake (amount of coins).
- Less energy-intensive than PoW.
- Examples: Ethereum 2.0, Cardano.

4. **Proof of Burn**
- Users burn coins (send to irretrievable address) to gain mining rights.
- Creates scarcity, encouraging long-term commitment.

5. **Difficulty Level**
- Adjusted to maintain block creation time.
- Higher difficulty means more computation required for PoW.

6. **Sybil Attack**
- A malicious user creates multiple fake identities to gain influence.
- PoW/PoS helps resist these attacks.

7. **Energy Utilization**
- Major concern with PoW (e.g., Bitcoin).
- PoS and alternative methods aim to reduce energy usage.

8. **Alternate Consensus Mechanisms**
- **Delegated Proof of Stake (DPoS)**: Users vote for delegates to validate blocks (e.g., EOS).
- **Practical Byzantine Fault Tolerance (PBFT)**: Tolerates faulty nodes, used in Hyperledger.
- **Proof of Authority (PoA)**: Identity-based consensus, used in private blockchains.

Let me know when you're ready for **Module 4: Cryptocurrency**.

Here are detailed notes for **Module 4: Cryptocurrency**:

Module 4: Cryptocurrency

1. **Currency**

- Digital or virtual currency using cryptography for security.
- Operates independently of central banks.
- Example: Bitcoin, Ethereum, Litecoin.

2. **Double Spending**

- Risk of spending the same digital currency twice.
- Solved using consensus mechanisms and transaction history.

3. **Cryptocurrency**

- A type of digital currency built on blockchain technology.
- Secured by cryptographic techniques.
- Enables peer-to-peer transfers without intermediaries.

4. **P2P Payment Gateway**

- Peer-to-peer (P2P) payment allows users to send/receive money directly.
- No need for a central clearing authority.
- Examples: Bitcoin, Lightning Network.

5. **Wallet**

- Digital application or device to store cryptocurrency.
- **Hot Wallets**: Connected to the internet (e.g., mobile apps).
- **Cold Wallets**: Offline storage (e.g., hardware wallets).

6. **Mining**

- Process of validating transactions and adding them to the blockchain.
- Requires solving complex puzzles (in PoW).
- Miners are rewarded with newly created cryptocurrency.

7. **Public Blockchain and Private Blockchain**

- **Public**: Open participation, decentralized, e.g., Bitcoin, Ethereum.

- **Private**: Restricted access, used by organizations, e.g., Hyperledger.

8. **Other Cryptocurrencies**

- **Ethereum (ETH)**: Smart contracts, dApps.
- **Ripple (XRP)**: Cross-border payments.
- **Litecoin (LTC)**: Faster transaction times than Bitcoin.
- **Monero (XMR)**: Privacy-focused transactions.

Let me know when to proceed with **Module 5: Ethereum and Hyperledger Fabric**.

Here are detailed notes for **Module 5: Ethereum and Hyperledger Fabric**:

Module 5: Ethereum and Hyperledger Fabric

1. **Ethereum Network**

- Open-source blockchain with smart contract functionality.
- Introduced by Vitalik Buterin in 2015.
- Used for dApps, DeFi, NFTs.

2. **EVM (Ethereum Virtual Machine)**

- Runtime environment for smart contracts.
- Executes bytecode on all nodes for consensus.
- Platform-independent and secure.

3. **Transaction Fee, Gas, Ether**

- **Gas**: Unit of computation work.
- **Ether (ETH)**: Native cryptocurrency used to pay gas.
- More complex operations require more gas.

4. **Mist, Web3, Truffle**

- **Mist**: Ethereum's deprecated official wallet.

- **Web3.js**: JavaScript library to interact with Ethereum.
- **Truffle**: Development framework for testing and deploying smart contracts.

5. Solidity

- Contract-oriented programming language for writing smart contracts.
- Syntax similar to JavaScript.

6. Smart Contracts

- Self-executing contracts with predefined rules.
- Stored and run on the blockchain.
- Cannot be altered once deployed.

7. Design and Issue Cryptocurrency

- Ethereum enables users to create ERC-20 or ERC-721 tokens.
- Used in ICOs and NFTs.

8. Introduction to Hyperledger

- Open-source collaborative project by Linux Foundation.
- Provides enterprise-level blockchain frameworks.

9. Hyperledger Fabric Architecture

- Modular architecture with plug-and-play components.
- Uses channels for private transactions.
- Roles: Peers, Orderers, Certificate Authorities.

10. Features of Hyperledger Fabric

- Permissioned network with identity management.
- Supports private data sharing.
- Chaincode (smart contracts) can be written in Go, JavaScript.

Let me know when you're ready for the final **Module 6: Applications of Blockchain**.

Here are detailed notes for **Module 6: Applications of Blockchain**:

Module 6: Applications of Blockchain

1. **Blockchain Use-Cases**

- Enhances transparency, security, and traceability across sectors.
- Reduces dependency on central authorities.

2. **Supply Chain Management**

- Tracks products from origin to delivery.
- Improves accountability and reduces fraud.
- Example: IBM Food Trust with Walmart.

3. **Healthcare**

- Secures patient data and medical records.
- Ensures data integrity and accessibility.
- Enables interoperable health information exchange.

4. **Finance and Banking**

- Facilitates faster cross-border payments.
- Reduces transaction costs and processing time.
- Examples: Ripple, JPM Coin.

5. **Voting Systems**

- Provides a tamper-proof and transparent election system.
- Enhances voter anonymity and verifiability.

6. **Digital Identity**

- Creates self-sovereign identity systems.
- Users control their data without relying on intermediaries.
- Example: uPort, Sovrin.

7. **Real Estate**

- Tokenizes property assets for easier transfer and investment.
- Smart contracts automate leasing and ownership transfer.

8. Government Services

- Reduces corruption and enhances transparency.
- Examples: Land records, subsidies, public distributions on blockchain.

9. Education

- Verifiable and immutable records of degrees and certifications.
- Example: MIT issuing diplomas on blockchain.

10. Other Areas

- **Media**: Protects copyright and ensures fair royalty distribution.
- **Gaming**: Ownership of in-game assets via NFTs.
- **Energy**: Peer-to-peer energy trading using blockchain.

You're now fully covered for all six modules. Let me know if you want revision pointers or a one-pager cheat sheet before your exam.