

BCT - UNIT 5 (Blockchain Ethereum Platform using Solidity) – END-SEM PYQ Answers➤ **MAY / JUN 2023****Q5) a) Discuss what is Ethereum. [8]**

Ethereum is an **open-source, blockchain-based platform** that allows developers to build and deploy decentralized applications (DApps) using smart contracts. It was proposed by **Vitalik Buterin in 2013** and launched in 2015 as an advancement over Bitcoin's limited scripting functionality.

- Ethereum extends blockchain technology beyond cryptocurrency by supporting programmable logic on its network.
- Ethereum's core component is the **Ethereum Virtual Machine (EVM)**, which executes smart contracts securely across all nodes in the network.
- The native cryptocurrency, **Ether (ETH)**, is used to pay for computational tasks and transaction fees known as **gas**. This ensures that resources are used efficiently and prevents network abuse.
- Unlike centralized systems, Ethereum operates in a **decentralized and transparent** manner where no single authority controls data or transactions. It supports the creation of tokens, DeFi (Decentralized Finance) applications, NFTs, and DAOs (Decentralized Autonomous Organizations).

Types of Ethereum Networks:**1. Ethereum Mainnet (Main Network):**

- This is the **primary public Ethereum blockchain** where real transactions occur using actual **Ether (ETH)**.

2. Ethereum Testnets (Testing Networks):

- These are used by developers to **test smart contracts or DApps** before deploying them on the mainnet.

3. Private Ethereum Network:

- A **custom network** set up by organizations or developers for internal use or experimentation.

Ethereum also enables **interoperability between multiple blockchain-based services** and promotes **ensorship-resistant applications**, ensuring that data remains secure and immutable. Its continuous upgrades, such as the shift from Proof of Work to **Proof of Stake (Ethereum 2.0)**, improve scalability, efficiency, and energy consumption.

b) Explain in detail Whisper. [6]

Whisper is a decentralized messaging protocol developed as part of the Ethereum ecosystem to enable secure, peer-to-peer communication between DApps and users. It focuses on privacy and anonymity while ensuring that messages are resistant to censorship.

Key Points:

- **Purpose:** Used to send encrypted messages between DApps running on Ethereum, enabling private communication.
- **Functionality:** Works on a peer-to-peer model, where each message is broadcasted to all nodes, ensuring no central point of control.
- **Encryption:** Whisper uses asymmetric encryption to secure message content, allowing only the intended receiver to decrypt it.
- **Anonymity:** Sender and receiver identities are hidden, ensuring complete privacy during communication.
- **Use Case:** Commonly used in decentralized chat systems, voting applications, and notifications in DApps.
- **Integration:** Whisper works alongside Swarm (for storage) and Ethereum blockchain (for computation) to complete the Ethereum web 3.0 stack.

It plays a vital role in providing privacy-preserving communication for decentralized ecosystems.

c) Define the purpose and types of Smart Contracts. [4]

Smart contracts are self-executing digital agreements coded on the blockchain, where the terms of the contract are directly written into code. They automatically execute actions when predefined conditions are met.

Purpose:

- To automate transactions without intermediaries.
- To ensure transparency, security, and trust in digital agreements.
- To reduce time and cost in business processes by removing manual verification.

Types of Smart Contracts:

1. **Deterministic Smart Contracts:** Execute automatically when specific conditions are met.
2. **Non-deterministic Smart Contracts:** Depend on external data from oracles or APIs.
3. **Smart Legal Contracts:** Legally binding agreements written partly in code and partly in natural language.

Smart contracts are the foundation of decentralized applications and form the backbone of modern blockchain-based systems.

Q6) a) Write a note on Ethereum Virtual Machine (EVM). [8]

The **Ethereum Virtual Machine (EVM)** is the core component of the Ethereum blockchain that executes smart contracts and manages the state of the entire network. It functions as a **decentralized, global computer** that ensures every node processes code in the same way, maintaining consensus across the blockchain.

Key Points:

- **Execution Environment:** EVM provides a runtime environment for executing smart contract code in a secure and isolated manner.
- **Language Support:** Smart contracts written in **Solidity** or other Ethereum-supported languages are compiled into **EVM bytecode** for execution.
- **Isolation:** Code running inside the EVM cannot access the underlying system, which ensures safety from malicious code or external interference.
- **Gas Mechanism:** Each operation in the EVM requires a specific amount of **Gas**, paid in Ether (ETH), to prevent infinite loops and misuse of network resources.
- **Deterministic Computation:** Every node executes the same code and produces the same result, maintaining consensus across the network.
- **Cross-Platform Compatibility:** The EVM ensures that smart contracts behave consistently on all Ethereum nodes, regardless of the underlying hardware or OS.

The EVM is the backbone of Ethereum's programmable capabilities, allowing developers to build decentralized applications securely and reliably.

b) Write a note on Swarm. [6]

Swarm is a decentralized storage platform developed as part of the Ethereum ecosystem. It allows distributed storage and sharing of data across the network without relying on centralized servers.

Key Points:

- **Purpose:** Designed to provide decentralized and censorship-resistant file storage for Ethereum-based applications.
- **Integration:** Works alongside Ethereum for computation and Whisper for communication, forming the complete **Web 3.0 infrastructure**.
- **Data Distribution:** Files are split into chunks and distributed across multiple nodes, improving data redundancy and reliability.
- **Incentive System:** Nodes earn rewards (using Ether or native tokens) for contributing their storage and bandwidth to the network.
- **Fault Tolerance:** Even if some nodes go offline, data can be reconstructed from other peers due to replication.
- **Use Case:** Commonly used for hosting decentralized websites, DApp data, and smart contract-related files.

Swarm ensures that data remains permanently available, secure, and free from centralized control — supporting the true vision of a decentralized web.

c) What is a Smart Contract? Explain with an example. [4]

A **Smart Contract** is a self-executing agreement stored on the blockchain, where the terms and rules are directly written into code. It automatically performs actions when specified conditions are met, without needing intermediaries.

Key Points:

- Executes automatically once pre-defined conditions are satisfied.
- Ensures transparency and eliminates the need for third-party verification.
- Reduces costs and errors through automation and immutability.

Example:

A simple **smart contract for crowdfunding** can automatically release funds to a project owner only if the funding goal is reached by a certain date. If not, it refunds the contributors automatically — all handled securely on the blockchain.

➤ MAY / JUN 2024

Q5) a) What is Ethereum? Explain in detail how it differs from traditional cryptocurrencies. [8]

Ethereum is an open-source, decentralized blockchain platform that allows developers to build and deploy **smart contracts** and **decentralized applications (DApps)**. It extends the concept of blockchain beyond digital currency, enabling programmable transactions and automation through its own scripting language, **Solidity**.

Key Points:

- **Purpose:** Ethereum enables not only peer-to-peer payments but also programmable agreements that execute automatically.
- **Native Currency:** It uses **Ether (ETH)** as its cryptocurrency, which powers transactions and smart contract operations.
- **Ethereum Virtual Machine (EVM):** Acts as a global decentralized computer executing contract code securely across nodes.
- **Smart Contracts:** Allow developers to automate processes without intermediaries, increasing efficiency and reducing cost.
- **Decentralized Applications (DApps):** Used in areas such as DeFi, gaming, healthcare, and logistics.

Differences from Traditional Cryptocurrencies:

- Traditional cryptocurrencies like **Bitcoin** are designed primarily for peer-to-peer digital payments.
- Ethereum, on the other hand, is a **platform for building decentralized applications**, not just a currency.

- Bitcoin transactions are fixed and simple, whereas Ethereum supports **programmable logic** via smart contracts.
- Ethereum has a faster block time and supports **token creation (ERC-20, ERC-721)**, while Bitcoin does not.
- Ethereum moved from **Proof of Work to Proof of Stake**, making it more energy-efficient than traditional cryptocurrencies.

Hence, Ethereum represents the evolution of blockchain from simple digital money to a programmable, decentralized ecosystem.

b) Identify two types of Ethereum networks and briefly explain the purpose of each. [9]

Ethereum provides multiple network environments that serve different roles in the **development, testing, and deployment** of decentralized applications (DApps). The two main types are the **Ethereum Mainnet** and the **Ethereum Testnets**, both essential for the smooth functioning of the blockchain ecosystem.

1. Ethereum Mainnet:

- The **Mainnet** is the **official public Ethereum blockchain** where all real transactions occur using **actual Ether (ETH)**.
- It represents the **production environment**, meaning that once a transaction or smart contract is deployed here, it becomes permanent and immutable.
- Developers deploy **final versions** of smart contracts and DApps on Mainnet for public use.
- Every transaction requires **gas fees** in real ETH to compensate miners or validators for computation.
- **Security and reliability** are highest here since thousands of nodes validate and secure the network.
- **Example:** Decentralized applications like **Uniswap, OpenSea, and Aave** operate on the Ethereum Mainnet.
- The Mainnet acts as the foundation of the Ethereum ecosystem where real-world value is transferred and decentralized apps function in a live environment.

2. Ethereum Testnets:

- **Testnets** are **experimental versions** of the Ethereum network used for testing and development.
- They replicate the same features as the Mainnet but use **test Ether**, which has **no monetary value**, ensuring a safe testing environment.
- Developers use Testnets to **deploy, debug, and test** their smart contracts before launching on the Mainnet.
- It helps prevent loss of real Ether and avoids costly errors in production.

- Popular test networks include **Goerli**, **Sepolia**, and **Rinkeby**.
- **Goerli** and **Sepolia** are currently active and used for Ethereum 2.0 and Proof of Stake testing.
- Testnets allow developers to experiment freely, conduct community testing, and optimize their DApps for real-world deployment.

Together, the **Ethereum Mainnet and Testnets** form a complete ecosystem that supports both safe experimentation and real-world blockchain operations.

This dual-network model ensures reliability, scalability, and continuous innovation in the Ethereum environment.

Q6) a) Explain the concept of a smart contract with an example. [8]

A **Smart Contract** is a self-executing digital agreement stored on the blockchain, where the terms of the contract are written directly in code. These contracts automatically perform and enforce actions when certain predefined conditions are met, eliminating the need for intermediaries such as banks, lawyers, or brokers.

Key Points:

- Smart contracts run on blockchain platforms like **Ethereum**, ensuring transparency, immutability, and automation.
- Once deployed, they cannot be altered — making them **tamper-proof** and **trustless**.
- Transactions executed through smart contracts are **recorded permanently** on the blockchain, providing full traceability.
- They significantly **reduce costs, delays, and human errors** compared to traditional paper-based contracts.
- Developers generally use programming languages such as **Solidity** to create and deploy them on the Ethereum Virtual Machine (EVM).

Example:

Consider a **rental agreement** between a tenant and a landlord coded into a smart contract.

- The tenant pays rent in cryptocurrency to the contract.
- Once payment is received, the smart contract automatically transfers access to the digital lock of the apartment.
- If the rent isn't paid by the due date, the contract automatically denies access until payment is made.

Advantages and Impact:

- Ensures **automatic execution** without third-party intervention.
- Guarantees **security** since data is encrypted and distributed across nodes.
- Promotes **trust and transparency** between parties through verifiable code.

Thus, smart contracts bring automation, fairness, and efficiency to digital agreements, revolutionizing how transactions and legal processes are executed in the blockchain world

b) How does Swarm contribute to decentralized storage in the Ethereum ecosystem? [9]

Swarm is a **decentralized storage and content distribution system** that forms an integral part of the Ethereum ecosystem.

It is designed to provide **peer-to-peer (P2P) storage and sharing of data** without relying on centralized servers.

- Swarm ensures that data and DApps built on Ethereum can store, access, and distribute content in a decentralized and censorship-resistant way.
- Swarm contributes to decentralized storage by allowing users to **upload files as chunks** that are distributed across various network nodes. Each node stores small parts of data, and redundancy ensures availability even if some nodes go offline.
- The system uses **content addressing and hashing**, meaning files are retrieved using their hash rather than a traditional URL, ensuring data integrity and immutability.
- It also uses **incentivization mechanisms through Ethereum-based payment systems**, rewarding nodes that store and deliver data reliably.
- This aligns with Ethereum's vision of a fully decentralized web, known as **Web3**, where users own and control their data.
- Additionally, Swarm integrates seamlessly with smart contracts and DApps, enabling developers to build applications that store documents, media, and blockchain-related data in a distributed manner.

In summary, **Swarm enhances Ethereum by providing reliable, censorship-free, and scalable decentralized storage**, reducing dependence on centralized servers while maintaining transparency and security across the blockchain network. It supports the long-term sustainability of decentralized applications and digital assets within the Ethereum ecosystem

➤ MAY / JUN 2025

Q5) a) Enlist features of Ethereum. Compare Ethereum Mainnet, Testnet, and Private Networks. [9]

Features of Ethereum:

1. **Smart Contracts:** Ethereum allows automated agreements that execute when specific conditions are met, reducing the need for intermediaries.
2. **Decentralization:** Operates on a distributed network of nodes, ensuring transparency and resistance to censorship.
3. **Ether (ETH):** The native cryptocurrency used to pay transaction fees and reward miners or validators.
4. **EVM (Ethereum Virtual Machine):** A runtime environment that enables the execution of smart contracts across all nodes.
5. **DApps Support:** Developers can build and deploy decentralized applications using Solidity on Ethereum.
6. **Security:** Uses cryptographic algorithms to ensure secure transactions and contract execution.

7. **Open Source:** Ethereum's source code is publicly available, promoting transparency and innovation.
8. **Gas Mechanism:** Introduces gas to manage computational costs and prevent misuse of network resources.
9. **Upgradability:** Ethereum supports network upgrades and protocol improvements through hard or soft forks.

Comparison between Ethereum Mainnet, Testnet, and Private Networks:

Aspect	Ethereum Mainnet	Testnet	Private Network
Purpose	The live public network used for real transactions.	Used by developers for testing and experimentation.	Used within organizations or individuals for private development.
Currency	Uses real Ether (ETH) with actual monetary value.	Uses test Ether (free, without real value).	Uses custom or test Ether with no real-world value.
Accessibility	Fully public and decentralized.	Public but separate from the main blockchain.	Restricted access, controlled by organization.
Cost	Transactions require real ETH (gas fees).	No real cost; ETH is freely available via faucets.	No real cost; controlled environment.
Use Case	Real DApps and production-level smart contracts.	Testing DApps, smart contracts, or protocol updates.	Experimentation, internal testing, or educational use.
Security	High security as it's supported by global validators.	Lower security as it's for testing purposes.	Depends on organization's setup and permissions.
Consensus Mechanism	Proof of Stake (PoS).	Usually mirrors Mainnet's mechanism.	Can use PoS, PoW, or custom protocols.
Examples	Ethereum Mainnet	Goerli, Sepolia Testnets	Enterprise or academic Ethereum networks

b) Define smart contracts with an example of a real-world scenario which will evaluate its benefits. Discuss the role of gas in the Ethereum network and its significance in smart contract execution.
[9]

Smart Contracts:

A **smart contract** is a self-executing program stored on the blockchain that automatically enforces the terms of an agreement once the predefined conditions are met. These contracts eliminate the need for intermediaries and ensure trust through transparency and immutability.

Example (Real-world Scenario):

Consider an **insurance claim system** — when a flight gets canceled, the smart contract verifies flight data through an oracle and automatically releases the insurance payment to the passenger without manual verification.

Benefits:

- **Transparency:** All participants can view the contract terms.
- **Automation:** Executes automatically, reducing human intervention.
- **Cost Reduction:** Eliminates intermediaries, saving time and fees.
- **Security:** Data is immutable and stored permanently on the blockchain.
- **Trustless Transactions:** Parties need not trust each other, only the blockchain.

Role of Gas in Ethereum Network:

- **Definition:** Gas is the **unit of computational work** required to execute operations like transactions or smart contracts on Ethereum.
- **Gas Fees:** Users pay gas fees in Ether (ETH) to compensate miners or validators for the resources used.
- **Purpose:**
 - Prevents network abuse by making spam transactions costly.
 - Measures computational effort for each operation.
 - Rewards validators who confirm transactions.
- **Significance in Smart Contracts:**
 - Complex contracts consume more gas depending on the number of operations executed.
 - Gas ensures that infinite loops or poorly written contracts don't congest the network.
 - Developers must optimize smart contract code to minimize gas consumption.

Smart contracts revolutionize digital agreements by enabling secure, transparent, and automated execution, while the **gas mechanism** ensures fair network operation, efficiency, and sustainability of the Ethereum ecosystem.

Q6) a) List Decentralized Messaging Platforms. Discuss Whisper and its purpose in the Ethereum ecosystem. [9]

Decentralized Messaging Platforms:

In the Ethereum ecosystem, decentralized messaging platforms enable **secure, peer-to-peer communication** without relying on centralized servers. The most recognized platforms are:

1. **Whisper** – Ethereum’s native decentralized messaging protocol.
2. **Status** – Built on Whisper to provide secure mobile messaging integrated with Web3.
3. **Matrix** – An open standard for decentralized communication, interoperable with blockchain systems.

Whisper in Ethereum Ecosystem:

Whisper is a **peer-to-peer decentralized messaging protocol** developed as part of the Ethereum project to allow users and decentralized applications (DApps) to exchange encrypted messages securely. It complements Ethereum’s other core components — **Swarm (storage)** and **Ethereum blockchain (transaction layer)** — by providing the **communication layer**.

Purpose of Whisper:

1. **Private Communication:** Whisper enables users or DApps to communicate securely and anonymously using encryption.
2. **Off-chain Messaging:** It allows information exchange outside the main Ethereum chain, reducing blockchain load.
3. **Decentralization:** Messages are distributed across the network without any central server or control point.
4. **Temporary Message Storage:** Nodes can store messages temporarily, ensuring delivery even if recipients are offline.
5. **Integration with DApps:** Ideal for DApps requiring private messaging, notifications, or data exchange.

Example (Use Case):

In a decentralized auction DApp, Whisper can send real-time bid notifications securely to all participants without revealing their identities or relying on a central server.

Whisper plays a crucial role in enabling **secure, anonymous, and decentralized communication** in Ethereum. It ensures that privacy and peer-to-peer interaction remain integral parts of the blockchain ecosystem, supporting the foundation of Web3 communication.

b) Discuss in brief two types of smart contracts commonly used on the Ethereum platform. Elaborate the basic steps involved in deploying a smart contract on the Ethereum blockchain. [9]

Smart contracts are **self-executing programs** stored on the Ethereum blockchain that automatically perform actions when predefined conditions are met.

They eliminate intermediaries, reduce cost, and increase trust in digital transactions. On Ethereum, different types of smart contracts are used to serve various business or application needs.

Types of Smart Contracts:**1. Smart Legal Contracts:**

These contracts function like traditional legal agreements but exist digitally as code on the blockchain. They execute terms automatically once both parties fulfill their conditions.

Example: A rental contract that automatically transfers rent to the landlord once the tenant's digital signature and payment are verified. This ensures reliability and avoids legal disputes.

2. Decentralized Autonomous Organization (DAO) Contracts:

DAO contracts allow groups or organizations to make decisions collectively without a central authority. The entire governance process — from proposals to voting — is encoded in smart contracts.

Example: An investment DAO where members can vote on projects; funds are automatically distributed only if a proposal reaches the required votes. This makes operations fully transparent and secure.

3. Application Logic Contracts (ALC) – Additional Category:

These contracts define the specific logic behind decentralized applications (DApps), such as identity verification, asset tracking, or automated data handling.

They act as the “brain” of a DApp, ensuring that all internal processes run according to rules coded on the blockchain.

Steps Involved in Deploying a Smart Contract on Ethereum:

Step	Description
1. Writing the Contract	Developers use Solidity , Ethereum's contract language, to define functions, conditions, and events.
2. Compiling	The code is compiled using the Solidity compiler (solc) to generate bytecode and ABI (Application Binary Interface) for deployment.
3. Connecting to the Ethereum Network	Tools like MetaMask , Infura , or Alchemy connect developers to Ethereum's Mainnet or Testnet .
4. Deploying the Contract	Using Remix IDE , Truffle , or Hardhat , the contract is uploaded onto the blockchain. This step requires a gas fee paid in Ether to confirm the transaction.
5. Verification and Interaction	After deployment, the contract can be verified on Etherscan and interacted with using Web3.js or a DApp's frontend interface.

Smart contracts have truly **redefined trust and automation** in digital systems. By removing middlemen and encoding agreements directly into blockchain, they make processes faster, cheaper, and tamper-proof. The deployment cycle — from coding to verification — shows how Ethereum provides a complete framework for building decentralized applications that run exactly as programmed.

NOTE: Confirm the types of smart contracts mentioned here once!

➤ NOV / DEC 2023

Q5) a) What is Ethereum? What are the types of Ethereum? [6]

Ethereum is an open-source, blockchain-based platform that enables developers to build and deploy decentralized applications (DApps) using **smart contracts**. It extends the concept of Bitcoin by supporting programmable transactions beyond simple currency exchange.

Key Points:

- Ethereum uses its native cryptocurrency called **Ether (ETH)** for transaction fees and computational services.
- It allows decentralized finance (DeFi), NFT trading, and many blockchain-based projects.
- The platform operates using the **Ethereum Virtual Machine (EVM)**, which executes smart contract code.
- Transactions are secured using cryptographic algorithms and stored on the distributed ledger.

Types of Ethereum Networks:

1. **Mainnet:**
The live, public Ethereum network where real transactions occur and Ether has real value. It is the production environment for DApps and smart contracts.
2. **Testnet:**
Used by developers to test DApps and contracts without using real Ether.
Examples: **Goerli, Sepolia** — help simulate real blockchain activity safely.
3. **Private Network:**
Created for organizations to run Ethereum nodes internally.
Offers controlled access, faster performance, and data privacy for enterprise blockchain applications.

Ethereum thus serves as both a **cryptocurrency platform and a decentralized application ecosystem**, bridging financial, business, and technical innovation securely.

b) Explain SWARM (Decentralized Storage Platform) in detail. [6]

Swarm is a decentralized storage platform and content distribution service that forms an integral part of the Ethereum ecosystem.

It is designed to store and share data across a distributed network of nodes rather than on centralized servers. Each participant in the network contributes storage capacity and bandwidth, ensuring that data remains available and resistant to censorship or single points of failure.

Swarm operates using **peer-to-peer (P2P)** technology, where files are divided into small chunks and distributed among multiple nodes.

When users request a file, the chunks are reassembled automatically, ensuring fast and reliable access. This decentralized model eliminates dependency on cloud providers such as Google Drive or AWS.

Key Features:

- Provides **redundant and fault-tolerant data storage** through distributed nodes.
- Uses **Ethereum-based incentives** — participants earn rewards for sharing storage and bandwidth.
- Supports **immutability and persistence**, ensuring that stored data cannot be tampered with.
- Integrates smoothly with Ethereum smart contracts and DApps for storing documents, logs, or website content.
- Works closely with **Whisper** for messaging and **EVM** for computation — completing Ethereum’s “Web3” vision.

By decentralizing storage, Swarm ensures high data availability, privacy, and independence from centralized entities — making it a powerful backbone for decentralized web applications and blockchain-based file management.

c) What is the concept of Smart Contract? Discuss its types. [6]

A **Smart Contract** is a computer program stored on a blockchain that automatically executes or enforces the terms of an agreement once predefined conditions are met.

Unlike traditional contracts that rely on intermediaries such as banks or lawyers, smart contracts execute directly when conditions are satisfied, ensuring transparency, automation, and trust between unknown parties.

Smart contracts are written in languages like **Solidity** and are executed by the **Ethereum Virtual Machine (EVM)**. Each node in the network processes the same contract to reach the same outcome, making the system tamper-proof and decentralized. Once deployed, the code cannot be altered, which ensures the integrity of the agreement.

They are extensively used in **DeFi applications, NFTs, crowdfunding, and supply-chain automation**.

Types of Smart Contracts:

1. **Smart Legal Contracts:**
These contain legally enforceable terms between two or more parties. They can automate legal processes like digital payments, property transfers, or loan agreements. For example, a smart contract could automatically release payment when goods are confirmed as delivered.
2. **Decentralized Autonomous Organizations (DAOs):**
These are self-governing entities where decision-making rules are encoded within the contract. DAO members vote on proposals, and results are automatically enforced by the smart contract without any central authority.
3. **Application Logic Contracts (ALCs):**
These manage specific logic for decentralized applications and often connect off-chain

systems (like IoT devices) with blockchain networks. For instance, an IoT sensor could trigger a payment once a temperature threshold is reached.

Through automation, immutability, and transparency, smart contracts eliminate third-party involvement, reduce costs, and enhance the reliability of digital agreements — forming the **backbone of decentralized applications (DApps)** in the Ethereum ecosystem.

Q6) a) What is the significance of EVM (Ethereum Virtual Machine)? [6]

The **Ethereum Virtual Machine (EVM)** is the core component of the Ethereum blockchain that allows developers to create and execute **smart contracts** and **decentralized applications (DApps)**. It acts as a decentralized computing engine that ensures every transaction and contract runs exactly as programmed, without downtime, censorship, or third-party interference.

The EVM functions as a **runtime environment** for executing bytecode — the compiled version of smart contracts written in languages like **Solidity** or **Vyper**. Every Ethereum node runs its own instance of the EVM, ensuring that all nodes reach the same result for every computation, maintaining **consensus** across the blockchain network.

Significance of EVM:

1. **Platform Independence:**
The EVM provides a consistent environment for executing smart contracts across all nodes, making Ethereum applications platform-independent and universally compatible.
2. **Security and Isolation:**
Smart contracts run in an isolated environment (sandbox) within the EVM, preventing malicious code from affecting the rest of the system or the blockchain.
3. **Deterministic Execution:**
EVM ensures that smart contract outputs remain the same regardless of where or when they are executed, ensuring reliability and predictability in blockchain operations.
4. **Gas Mechanism:**
The EVM uses **gas** to measure computational effort required to execute transactions. This prevents infinite loops and ensures fair usage of network resources.
5. **Supports DApp Development:**
By offering an environment for executing complex logic, the EVM enables developers to build decentralized financial systems, games, marketplaces, and governance platforms.

Thus, the EVM serves as the **backbone of Ethereum's computational power**, providing a secure, decentralized, and universal framework for executing smart contracts and building Web3 applications efficiently.

b) Explain Whisper (Decentralized Messaging Platform) in detail. [6]

Whisper is a **peer-to-peer decentralized messaging protocol** developed as part of the Ethereum ecosystem. It enables secure, private, and anonymous communication between decentralized applications (DApps) and users. Whisper complements Ethereum's other core components — **Swarm**

for storage and **EVM** for computation — to complete the full Web3 vision of decentralized web infrastructure.

Whisper allows DApps to exchange information without exposing user identities or relying on centralized servers. Messages are **end-to-end encrypted** and distributed across a network of Ethereum nodes, ensuring privacy and resistance to censorship. It uses a **broadcast model**, where messages are sent to all nodes, but only the intended recipient can decrypt them using the correct key.

Key Features:

1. **Privacy and Anonymity:**
Whisper uses cryptographic encryption to ensure that messages remain confidential and sender identities are protected.
2. **Peer-to-Peer Communication:**
Operates without any central authority — messages are directly exchanged between nodes on the Ethereum network.
3. **Time-to-Live (TTL) and Topics:**
Each message can have a defined lifespan (TTL) and topic, which helps in filtering relevant messages efficiently.
4. **Integration with DApps:**
DApps use Whisper to notify users of updates, perform secure communication, and share temporary or sensitive data off-chain.
5. **Complementary Role:**
Works alongside Swarm (for data storage) and Ethereum blockchain (for computation), completing the decentralized ecosystem for Web3.

Example:

In a decentralized auction DApp, Whisper can be used to privately send bid amounts between users and the application without revealing them publicly on the blockchain.

Whisper thus plays a crucial role in enabling **secure, decentralized communication** within Ethereum-based applications, promoting privacy, freedom of speech, and trustless data exchange in the blockchain ecosystem.

c) Discuss how smart contracts are implemented using Solidity. [6]

Smart contracts are self-executing programs that run on the Ethereum blockchain when specific conditions are met. Solidity is the primary high-level programming language used to create, deploy, and manage these contracts. It is statically typed and influenced by languages like JavaScript and C++.

Implementation Steps using Solidity:

1. **Step 1 – Define the Contract:**
The contract is written in Solidity with a .sol extension. Developers define variables,

functions, and events.

Example:

```
solidity

pragma solidity ^0.8.0;
contract SimpleStorage {
    uint data;
    function set(uint x) public { data = x; }
    function get() public view returns (uint) { return data; }
}
```

2. **Step 2 – Compilation:**

The Solidity code is compiled using the **Solidity Compiler (solc)** which converts it into **EVM bytecode** and **ABI (Application Binary Interface)**. These are needed for deployment on the blockchain.

3. **Step 3 – Deployment:**

The compiled contract is deployed on the Ethereum network (Mainnet or Testnet) using tools like **Remix IDE**, **Truffle**, or **Hardhat**. During deployment, a small amount of **Ether (gas fee)** is paid to the network.

4. **Step 4 – Execution:**

Once deployed, users can interact with the contract through its address. The functions are executed in a decentralized manner by the Ethereum Virtual Machine (EVM), ensuring immutability and transparency.

5. **Step 5 – Verification and Testing:**

Before mainnet deployment, contracts are tested on **Testnets (like Ropsten or Goerli)** to check for errors, security vulnerabilities, and logical correctness.

Thus, Solidity provides a structured and secure way to implement smart contracts on Ethereum. It simplifies the creation of decentralized applications (DApps) by automating transactions without intermediaries, ensuring trust and efficiency in digital agreements.

➤ **NOV / DEC 2024**

Q5) a) What is Swarm, and how does it address the need for decentralized storage in the Ethereum network? [9]

Swarm is a **decentralized storage and content distribution system** that functions as the "hard drive" or storage layer of the Ethereum Web3 stack. It is a **peer-to-peer (P2P)** network designed to provide a censorship-resistant, fault-tolerant, and permissionless infrastructure for data, making it integral to the Ethereum ecosystem.

Key Role in Addressing Decentralized Storage: Swarm addresses the critical need for decentralized storage in the Ethereum network because the **Ethereum blockchain is not scalable or cost-effective**

for storing large data files (like documents, media, or heavy application assets). It provides an essential **off-chain solution** that complements the on-chain logic of smart contracts.

The mechanism by which Swarm achieves this decentralization involves several key points:

1. **Data Sharding and Distribution:** Swarm ensures decentralized storage by allowing users to upload files as small **chunks** that are distributed across various network **nodes** (called "Bee" nodes). Each node stores only small parts of the data, preventing any single point of failure.
2. **Redundancy and Availability:** Redundancy is built into the system, meaning multiple nodes store copies of the same chunks. This ensures **availability** and **fault tolerance** even if some nodes go offline.
3. **Content Addressing and Immutability:** The system uses **content addressing and hashing**. Files are retrieved using their unique cryptographic hash (based on the content) rather than a traditional location-based URL. This guarantees **data integrity** and **immutability**—if the content changes, the address changes.
4. **Incentivization Mechanism:** Swarm uses an economic model, leveraging **Ethereum-based payment systems** (using the BZZ token), to incentivize and reward nodes that reliably store and serve data. This ensures the long-term **persistence** and sustainability of the data.
5. **Smart Contract Integration:** Swarm integrates seamlessly with Ethereum **smart contracts** and DApps. Developers can build applications that store large data files in a distributed manner while using the efficient Ethereum blockchain only to store the **immutable content hash** (the pointer) for verification and retrieval.

In summary, Swarm enhances Ethereum by providing a **reliable, censorship-free, and scalable storage layer**, fulfilling Ethereum's vision of a fully decentralized Web3 where users maintain ownership and control over their data, reducing reliance on centralized servers.

b) What is Ethereum? Define smart contracts and give an example of a real-world scenario where a smart contract could be beneficial. [9]

Ethereum:

Ethereum is an **open-source, blockchain-based platform** that enables developers to build and deploy **decentralized applications (DApps)** using **smart contracts**. It was proposed by **Vitalik Buterin in 2013** and launched in 2015. Ethereum's **native cryptocurrency, Ether (ETH)**, is used to pay for transactions and computational services, known as **gas**, on the network. It provides a **decentralized, transparent, and trustless environment** where applications can operate without central authorities.

Smart Contracts:

A **smart contract** is a **self-executing program stored on the blockchain** that automatically performs predefined actions when certain conditions are met. They eliminate the need for intermediaries and ensure transparency, security, and immutability. Smart contracts are written in languages like **Solidity** and run on the **Ethereum Virtual Machine (EVM)**.

Example of a Real-World Scenario:

Consider **insurance claim processing**. A smart contract can automatically release payment to a policyholder if certain conditions, such as weather data indicating a flood or flight delay, are met. Once verified via a trusted oracle, the smart contract executes the payment without human intervention, reducing delays, errors, and administrative costs.

Benefits Highlighted:

- Reduces the need for intermediaries

- Ensures **automatic execution** and transparency
- Prevents fraud and tampering
- Increases efficiency in business processes

In summary, **Ethereum provides the platform** for deploying such trustless, automated smart contracts, making real-world processes faster, secure, and more reliable.

Q6) a) State and explain the various components of Ethereum. Explain the various types of Ethereum Networks. [9]

Components of Ethereum:

Ethereum is a decentralized blockchain platform with several key components that enable smart contracts and DApps to function efficiently:

1. Ethereum Virtual Machine (EVM):

- The **runtime environment** for smart contracts.
- Executes contract code in a **trustless and secure manner** across all nodes.
- Ensures all nodes reach the same results, maintaining **network consensus**.

2. Ether (ETH):

- The **native cryptocurrency** of Ethereum.
- Used as **gas** to pay for computations, smart contract executions, and transactions on the network.

3. Smart Contracts:

- **Self-executing programs** stored on the blockchain.
- Automatically perform actions when predefined conditions are met, eliminating intermediaries.
- Written in **Solidity** and executed on the EVM.

4. Accounts:

- Two types: **Externally Owned Accounts (EOA)** controlled by private keys and **Contract Accounts** that are controlled by smart contract code.
- EOAs initiate transactions; contract accounts execute logic automatically.

5. Nodes:

- Computers that maintain the Ethereum blockchain by validating transactions and smart contracts.
- Can be **full nodes** (store complete blockchain) or **light nodes** (store minimal data).

6. Gas:

- **Unit of computational cost** to prevent network abuse.
- Users pay gas fees in Ether to execute smart contracts and transactions.

Types of Ethereum Networks:

1. Ethereum Mainnet (Main Network):

- The **primary public network** where real Ether is used.
- Supports permanent deployment of smart contracts and DApps.

2. Ethereum Testnets (Testing Networks):

- Networks like **Sepolia, Goerli** are used for testing smart contracts **without using real Ether**.
- Developers can experiment safely before deploying on the mainnet.

3. Private Ethereum Network:

- **Custom, restricted network** set up by organizations for internal testing or enterprise solutions.
- Only authorized nodes can join; allows full control over mining, gas, and privacy.

b) Explain Decentralized Messaging Platform - Whisper and its purpose in the Ethereum ecosystem. [9]

Whisper is a **decentralized messaging protocol** designed to provide **secure, private, and anonymous communication** within the Ethereum ecosystem. It enables **peer-to-peer message exchange** between users and DApps without relying on any central server. Whisper complements Ethereum's **smart contracts** and **Swarm (decentralized storage)** to create a fully decentralized Web3 environment.

Key Features of Whisper:

1. Decentralized Communication:

- Messages are broadcast across the Ethereum network, and all nodes can relay messages.
- No single node has control over the network, ensuring a **trustless system**.

2. Privacy and Encryption:

- Whisper supports **end-to-end encryption**, keeping message contents confidential.
- Messages can be **ephemeral**, meaning they disappear after a set period, enhancing privacy.

3. Censorship Resistance:

- Messages cannot be easily blocked, altered, or traced, making it suitable for applications where **free communication** is critical.

4. Integration with DApps:

- DApps can use Whisper to send **notifications, alerts, or transaction updates** directly to users.
- Useful in scenarios like **decentralized voting, secure chat platforms, or financial alerts** in DeFi applications.

Purpose in the Ethereum Ecosystem:

- Provides a **communication layer** for DApps, complementing Ethereum's computation and storage layers.
- Ensures **secure and anonymous messaging**, maintaining decentralization principles.
- Supports **real-time signaling and notifications** between smart contracts and users.
- Helps Ethereum achieve its vision of **Web3**, where data, computation, storage, and communication are all **trustless and decentralized**.

Example:

In a **decentralized social network**, Whisper can be used to **send private messages between users** without any centralized server storing the data. This ensures privacy, censorship resistance, and direct peer-to-peer communication.

Note: Please check and verify all answers once before referring.