School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Dive into Ethereum Clients and EVM

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Ethereum Clients

• Ethereum Clients are specialized software applications that enable nodes to participate in the Ethereum blockchain network.
• They are responsible for connecting to peers, validating transactions and blocks, maintaining the blockchain state, and executing smart contracts.
• Different clients implement the Ethereum protocol using various programming languages — examples include **Geth (Go)**, **Nethermind (C#)**, **Besu (Java)**, and **Erigon (Go)**.
• The presence of multiple clients ensures decentralization, fault tolerance, and avoids single points of failure across the network.

Ethereum Virtual Machine (EVM)

• The Ethereum Virtual Machine is the core execution environment for running smart contracts on the Ethereum blockchain.
• It functions as a decentralized global computer that executes code in a secure and isolated environment.
• Smart contracts are written in high-level languages such as **Solidity** or **Vyper** and compiled into **EVM bytecode**, which every node executes to maintain consensus.
• The EVM ensures **deterministic execution**, meaning the same inputs always produce the same outputs across all nodes, ensuring network-wide consistency.
• It uses a **gas mechanism** to measure computational work, prevent resource abuse, and reward miners or validators for their processing power.
• The EVM's **isolation property** safeguards the network — if a contract fails or contains malicious code, it cannot affect other contracts or the blockchain itself.
• Through its robust architecture, the EVM supports the development of decentralized applications (DApps), decentralized finance (DeFi) systems, and NFT platforms, making Ethereum a powerful foundation for Web3 innovation.

* Softwares used

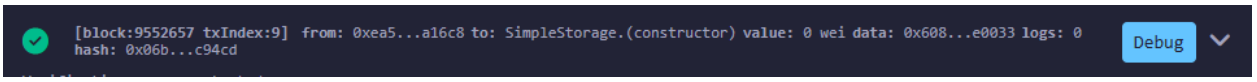
1. Brave browser
2. MetaMask Wallet
3. Remix IDE
4. Sepolia Testnet

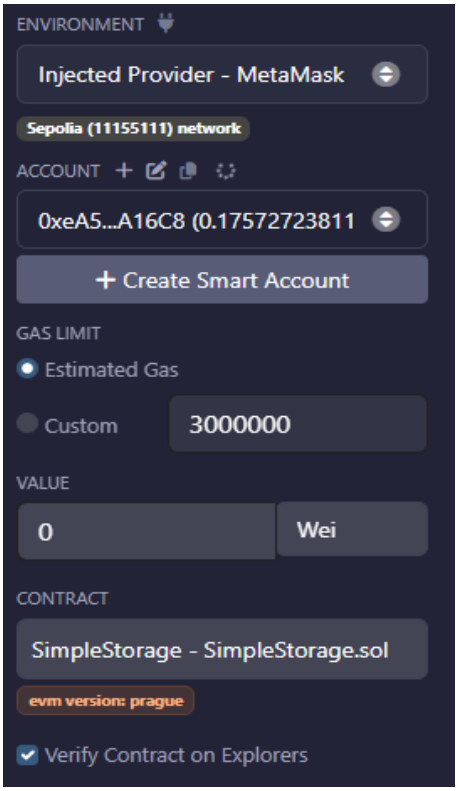
*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

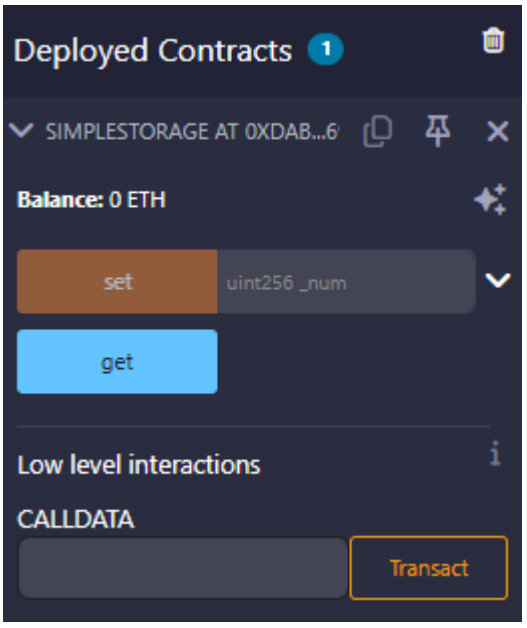
* Implementation Phase: Final Output (no error)

This Solidity program is a **simple storage smart contract** that lets users save a number on the blockchain and retrieve it later.
It contains one state variable (storedNumber), a setter function (setNumber) to update the value, and a getter function (getNumber) to read the value.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SimpleStorage {
    uint256 private storedNumber;

    // Function to store a number
    function set(uint256 _num) public {    22492 gas
        storedNumber = _num;
    }

    // Function to retrieve the number
    function get() public view returns (uint256) {    2431 gas
        return storedNumber;
    }
}
```







*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Observations:

- The Ethereum client was installed and configured successfully.
- The client synchronized with the Ethereum blockchain (testnet/mainnet as configured).
- Account creation and wallet address generation were successful.
- The client responded correctly to JSON-RPC/CLI commands.
- Transactions were executed and verified without errors.
- Logs/output confirmed the proper working of the Ethereum client.

The experiment was successfully carried out. The Ethereum client was installed, configured, and executed without errors. It synchronized with the blockchain, allowed account creation, and responded to commands correctly. This confirms that Ethereum clients are essential for interacting with the Ethereum network, validating transactions, and executing smart contracts through the EVM.

* Conclusion

The experiment was successfully carried out. The Ethereum client was installed, configured, and executed without errors. It synchronized with the blockchain, allowed account creation, and responded to commands correctly. This confirms that Ethereum clients are essential for interacting with the Ethereum network, validating transactions, and executing smart contracts through the EVM

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Regn. No. :

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.