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 :**

## * Coding Phase: Pseudo Code / Flow Chart / Algorithm

**Introduction:**

**Ethereum Clients**

- Software that allows nodes to connect, validate, and interact with the Ethereum network.
- Multiple clients exist (e.g., Geth, Nethermind, Besu, Erigon) — all implement Ethereum's protocol.
- Ensure decentralization and resilience by avoiding single-client dependency.
- Handle networking, consensus, and execution of transactions.

**Ethereum Virtual Machine (EVM)**

- The **execution environment** for Ethereum smart contracts.
- Acts as a global, decentralized "computer" that runs untrusted code securely.
- Executes bytecode compiled from smart contract languages (e.g., Solidity, Vyper).
- Maintains deterministic execution — same input always gives the same output across all nodes.
- Provides isolation, so buggy/malicious contracts can't affect the broader network.

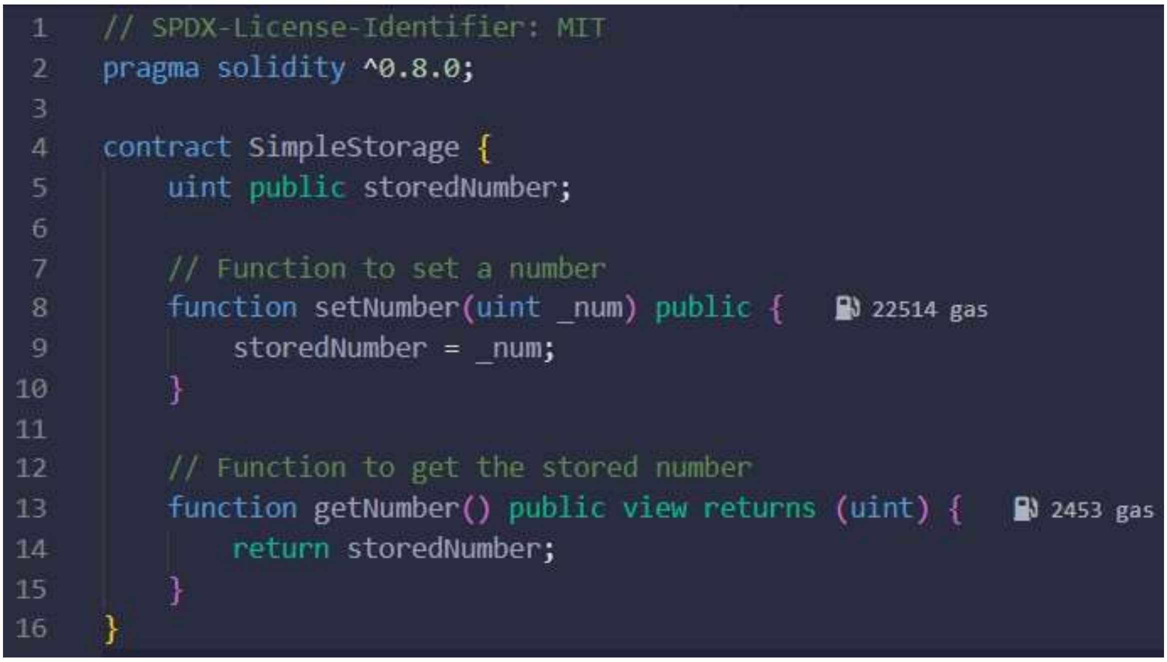
## * Softwares used

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. 🟩

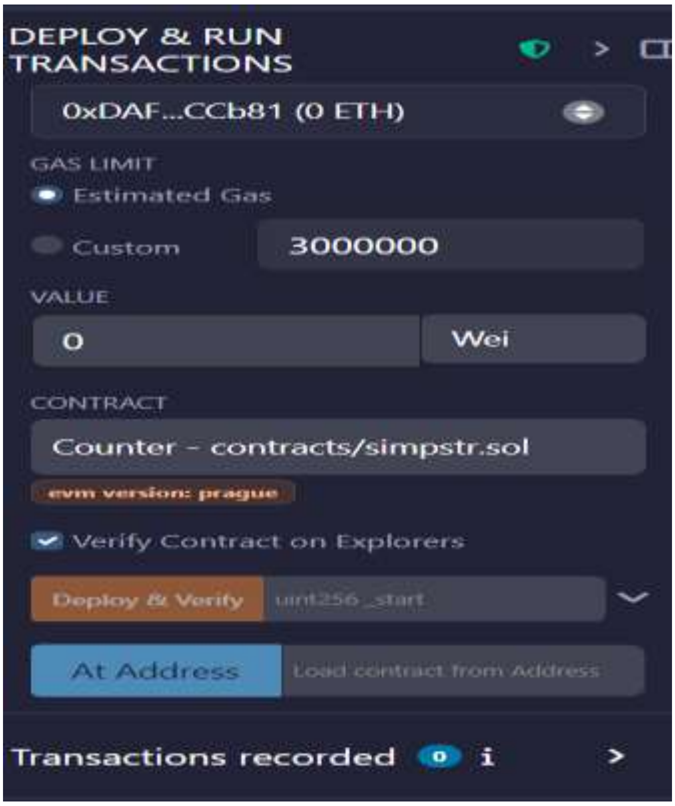


```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SimpleStorage {
    uint public storedNumber;

    // Function to set a number
    function setNumber(uint _num) public {     22514 gas
        storedNumber = _num;
    }

    // Function to get the stored number
    function getNumber() public view returns (uint) {     2453 gas
        return storedNumber;
    }
}
```



DEPLOY & RUN TRANSACTIONS

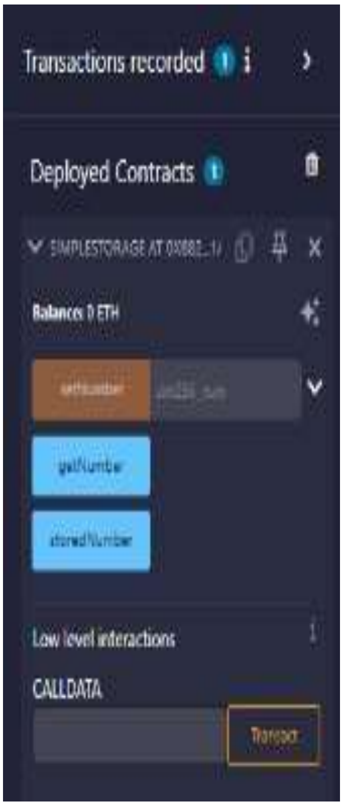
0xDAF...CCb81 (0 ETH)

GAS LIMIT
● Estimated Gas
○ Custom 3000000

VALUE
0 Wei

CONTRACT
Counter - contracts/simpstr.sol
evm version: prague

☑ Verify Contract on Explorers



Transactions recorded 1

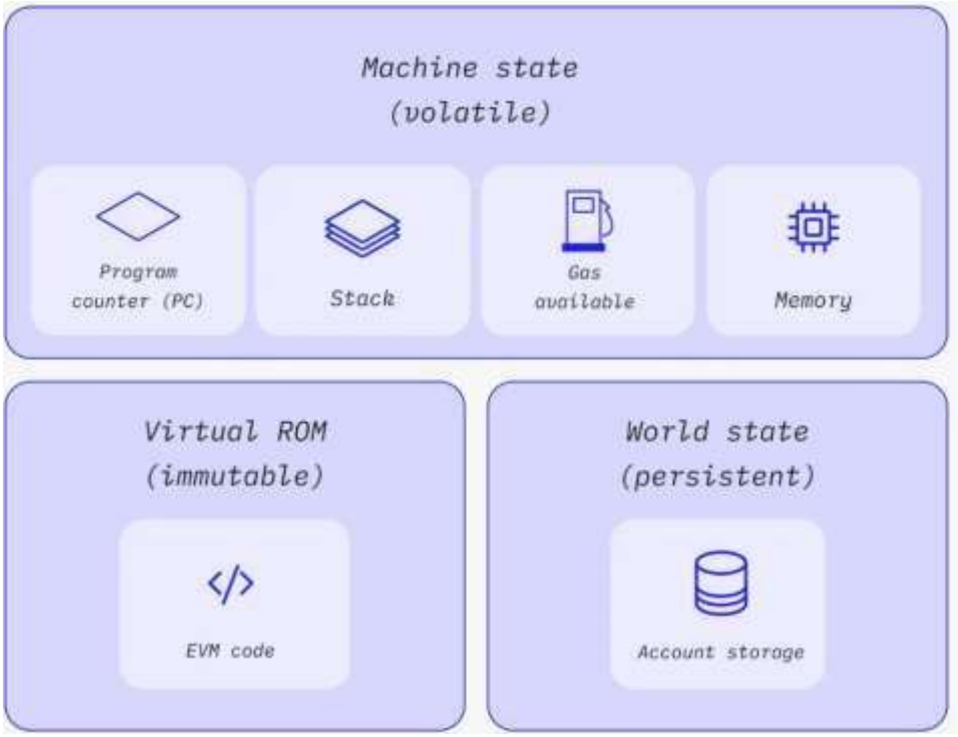
Deployed Contracts 1

SIMPLESTORAGE AT 0X882...1/

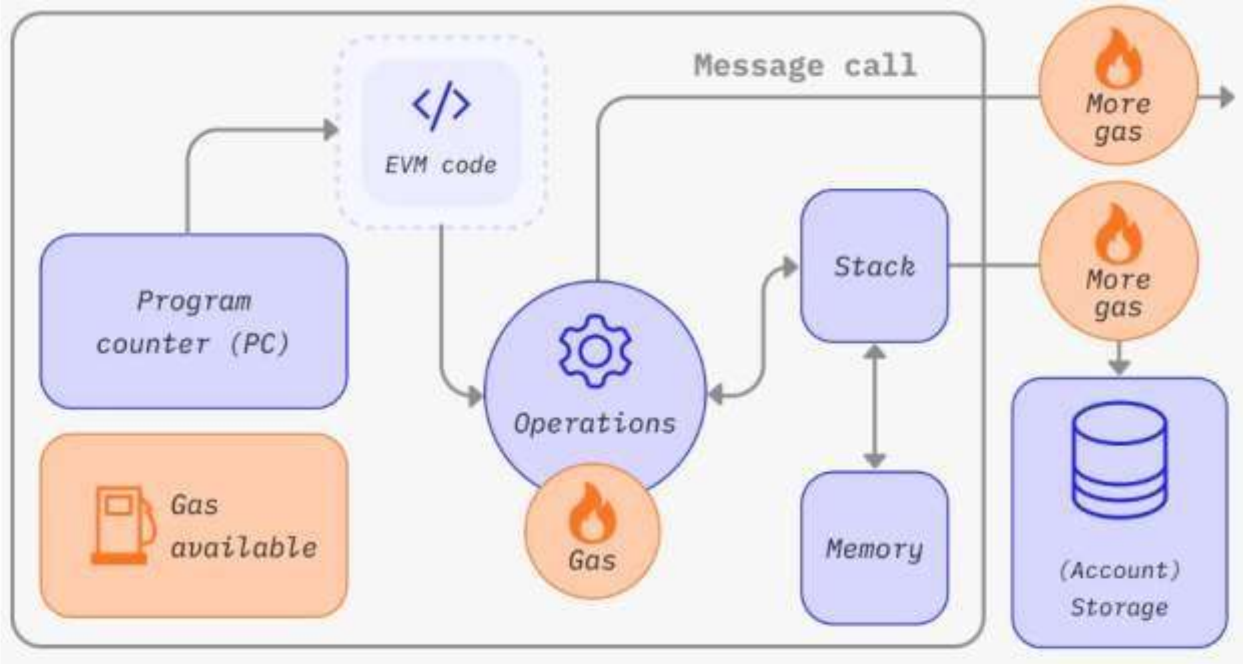
Balance: 0 ETH

## * Implementation Phase: Final Output (no error)

**Ethereum Virtual Machine (EVM)**



**EVM instructions**



*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/<br>Practical Simulation/ Programming | 10 | | |
| Result and Interpretation | 10 | | |
| Record of Applied and Action Learning | 10 | | |
| Viva | 10 | | |
| **Total** | **50** | | |

***Signature of the Student:***

***Signature of the faculty***:

*Name :*

*Regn. No. :*

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