

School:.............................................................................................................Campus:....................................................... AcademicYear:......................SubjectName:...........................................................SubjectCode:..........................

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

(LearningbyDoingandDiscovery)

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

**Introduction:**

Ethereum Clients

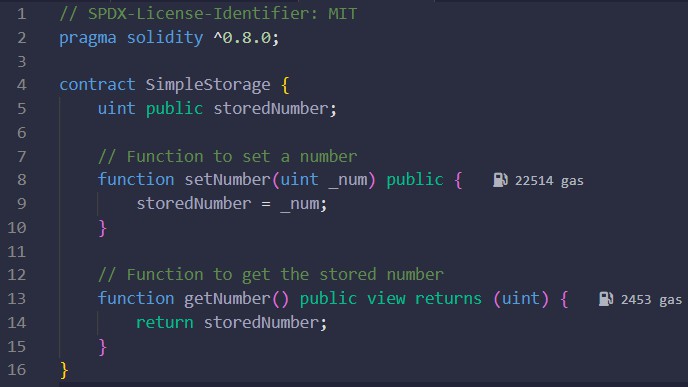
* Core software that lets computers (nodes) join the Ethereum blockchain.
* Examples: Geth (Go), Besu (Java), Nethermind (C#), Erigon (Rust).
* They maintain the ledger, validate transactions, and ensure consensus.
* Multiple implementations prevent centralization and improve security.

Ethereum Virtual Machine (EVM)

* The decentralized “engine” that runs smart contracts on all nodes.
* Translates high-level code (Solidity, Vyper) into bytecode.
* Guarantees that every node executes contracts in the same way.
* Provides a sandbox for contracts so errors or malicious actions can’t harm the blockchain.

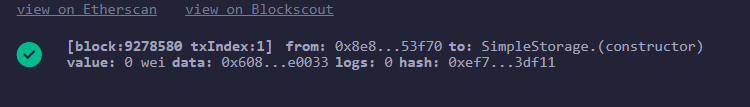
# \* Softwares used

* Google Chrome (with MetaMask extension)
* MetaMask Wallet for account and test ETH
* Remix IDE for writing & deploying smart contracts
* Go-Ethereum (Geth) client connected to Sepolia Testnet

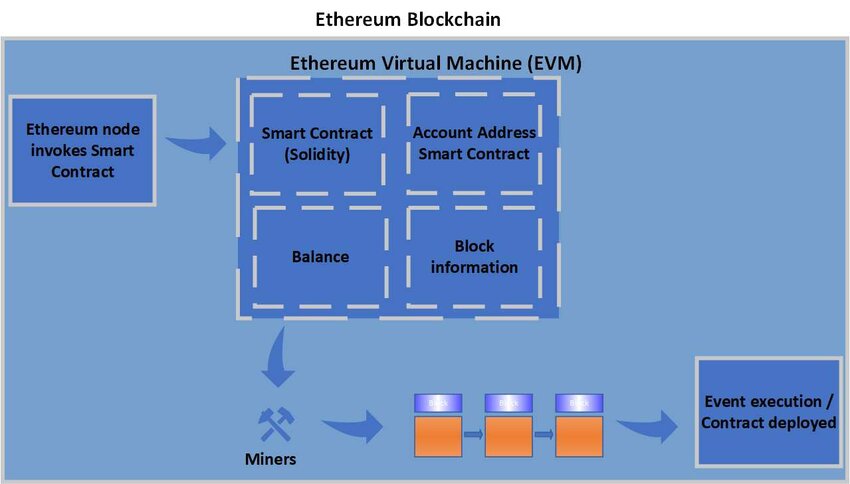


This Solidity program is a **simple storage smart contract** that lets users save a number on the blockchain and retrieve it later.

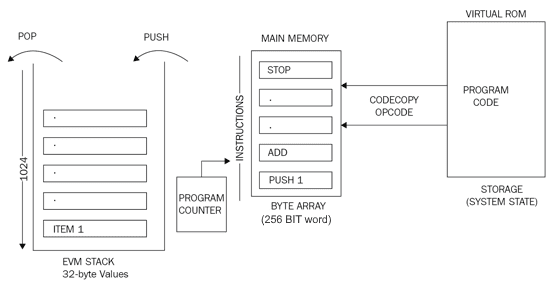
It contains one state variable (storedNumber), a setter function (setNumber) to update the value, and a getter function (getNumber) to read the value.



## Ethereum Virtual Machine (EVM)



**EVM instructions**



**\* Observations:** Applied and Action Learning



* + The Ethereum client was installed and successfully synced with Sepolia Testnet.
  + MetaMask generated an account and received test ETH.
  + The Counter contract was deployed using Remix IDE.
  + Increment transactions were executed, and values updated correctly in the EVM.
  + Blockchain logs showed consistent contract execution across nodes.

# \* Conclusion

The experiment confirmed how Ethereum clients maintain blockchain data and interact with the network, while the EVM ensures deterministic and secure execution of smart contracts. This demonstrates the practical role of Ethereum clients and the EVM in decentralized applications.



|  |  |  |  |
| --- | --- | --- | --- |
| **Rubrics** |  |  |  |
| 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:***

