



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 Experiment : Hello Solidity – Writing First Smart Contract

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM:

1. Start
2. Install the MetaMask extension in your browser and set up a wallet.
3. Connect to a test network (e.g., Sepolia) and collect test ETH from a faucet.
4. Open Remix IDE in your browser.
5. Create a new file named SimpleStorage.sol.
6. Write a simple contract in Solidity:
7. Write your code
8. Compile the contract using the Solidity Compiler in Remix.
9. In the Deploy tab, select Injected Provider - MetaMask and connect.
10. Click Deploy and confirm the transaction in MetaMask.
11. After deployment, click the greet() function to view the output.
12. End

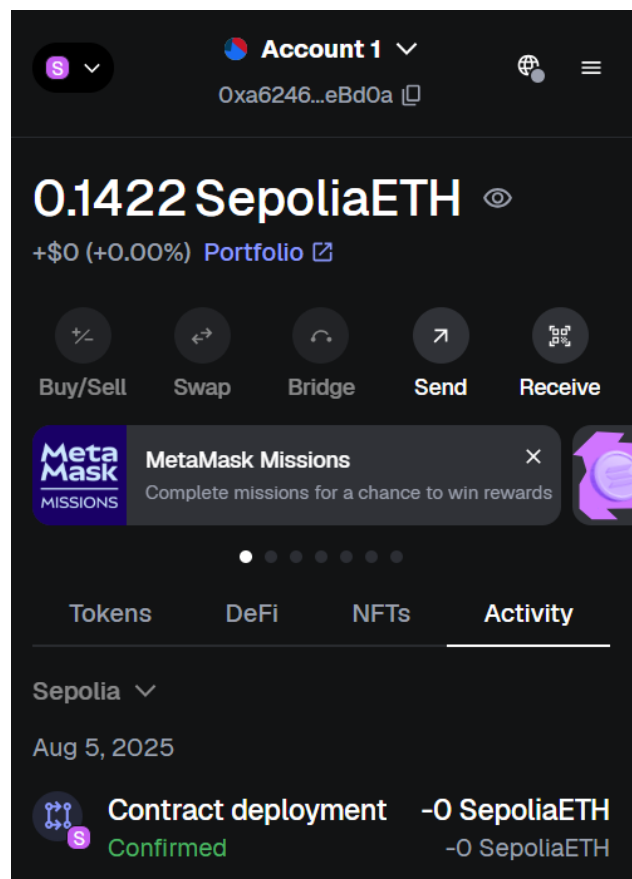
* Software Used:

1. Metamask Wallet
2. Remix IDE
3. Ethereum sepolia Faucet

* Testing/Implementation Phase:

First we have to download Metamask and add the extension on the brave browser where u can use the remix ide . Then go to Ethereum cloud to receive faucet





Give your wallet address to receive 0.05 faucet



Open remix IDE on the same browser and create a solidity file named as SimpleStorage.sol and write the code which we have to deploy

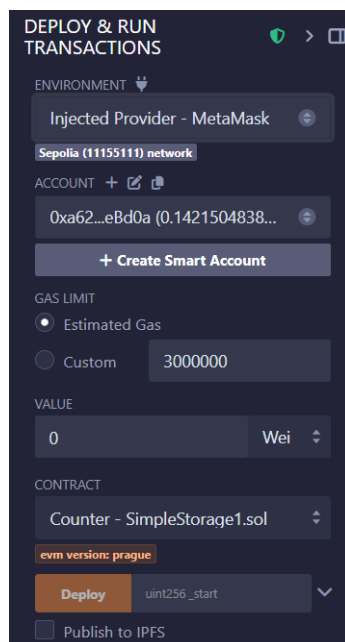
* Testing/Implementation Phase:)

```

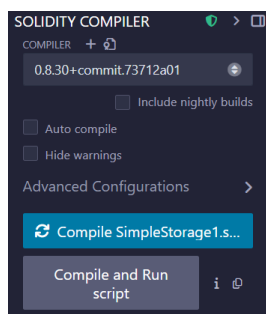
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.8.0;
3  contract Counter {
4      uint public count;
5      constructor(uint _start) {  infinite gas 132000 gas
6          count = _start;
7      }
8      function increment() public {  infinite gas
9          count += 1;
10     }
11     function decrement() public {  infinite gas
12         require(count > 0, "Counter is already at zero");
13         count -= 1;
14     }
15     function getCount() public view returns (uint) {  2453 gas
16         return count;
17     }
18 }

```

In code we have to create two function named get() and set() Get function to get the data and set function to set the data after write your smart contract go to the Environment and choose Injected Provider -Metamask and you can see there are autogenerate of your Metamask Wallet address with the test balance



Then click on compile after clicking our file has successfully compiled



* Implementation Phase: Final Output (no error)

Applied and Action Learning

Then click Deploy

```
Type the library name to see available commands.
creation of SimpleStorage pending...

view on Etherscan view on Blockscout

[block:8817273 txIndex:15] from: 0xe4a...ca52e to: SimpleStorage.constructor value: 0 wei data: 0x608...004d2 logs: 0 hash: 0x082...df851 Debug
```

Deployed Contracts 1

▼ SIMPLESTORAGE AT 0X69E...7

Balance: 0 ETH

set 234

get

storedData

0: uint256: 1234

Low level interactions

CALLDATA

Transact

```
call [call] from: 0xe4a0950de8a2c5c1646171858841668f8ca52e to: SimpleStorage.storedData() data: 0x2a1...afc09
transact to SimpleStorage.set pending ...

view on Etherscan view on Blockscout

[block:8817283 txIndex:35] from: 0xe4a...ca52e to: SimpleStorage.set(uint256) 0x69e...72fa3 value: 0 wei data: 0x60f...000ea logs: 0
hash: 0x5e3...ff2d2 Debug
```

Now Our smart contract has successfully deployed to our Wallet address

Sepolia

Jul 22, 2025

Set Confirmed -0 SepoliaETH

Contract deployment Confirmed -0 SepoliaETH

Set Confirmed -0 SepoliaETH

Contract deployment Confirmed -0 SepoliaETH

Observation:

1. The smart contract was successfully compiled and deployed on the Ethereum test network using Remix and MetaMask.
2. MetaMask handled the transaction and confirmed it on the blockchain.

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. :

Signature of the Faculty:

Page No.....

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