



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 :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Deploying Your First Smart Contract in Solidity on Remix IDE

1. Compiling the Code

- Remix automatically converts your Solidity contract into bytecode and ABI, so you don't need extra tools.
- It also checks that the code has no syntax errors and that you're using the correct compiler version.
- If the code has mistakes or the version doesn't match, deployment won't go through.

2. Connecting for Deployment

- MetaMask works as the link between Remix and the Ethereum blockchain (testnet or mainnet).
- When you deploy, MetaMask asks for confirmation and shows:
 - The wallet address that's deploying the contract
 - Estimated gas fees
 - The network being used
- This step shows that deployment happens via a blockchain transaction, not just by uploading the file.

3. Processing the Transaction

- You receive a transaction hash that you can follow on a block explorer like Etherscan or Sepolia Explorer.
- After confirmation, your contract is assigned a permanent address on the blockchain.
- Gas fees are deducted from your MetaMask account, proving that deploying contracts has a cost (even on testnets, it simulates real value).

4. Using the Contract

- Once live, Remix provides a built-in interface to interact with the contract functions.
- Calling functions that change data again needs MetaMask approval.
- You can see updates to the data or events directly in the Remix console.

Coding Phase: Pseudo Code / Flow Chart / Algorithm

Steps to Write and Deploy Your First Solidity Contract in Remix

Step 1: Open Remix IDE

- Go to <https://remix.ethereum.org> in your browser.

Step 2: Create a New File

- On the left panel, open the File Explorer.
- Inside the contracts folder, click the + (new file) button.
- Give your file a name, for example: SimpleStorage.sol.

Step 3: Write the Smart Contract

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

Step 4: Compile the Contract

- Click on the Solidity Compiler tab (the icon with a “tick mark”).
- Hit the Compile button.

Step 5: Deploy the Contract

- Switch to the Deploy & Run Transactions tab (Ethereum logo with play button).
- Under Environment, choose Injected Provider – MetaMask to connect your wallet. Click Deploy, and MetaMask will pop up to confirm the transaction.

Step 6: Interact with the Contract

Once deployed, you’ll see your contract under Deployed Contracts

Softwares used

- *Remix IDE*
- MetaMask Wallet

*** Testing Phase: Compilation of Code (error detection)**

No Error

* Implementation Phase: Final Output (no error)

The screenshot displays the Remix Online IDE interface. At the top, the Remix logo and text "Remix - Ethereum IDE & community" are visible, along with the note "Web site created using create-react-app". Below this, a section titled "Remix Online IDE" describes it as a "Web-based DevEnvironment" and a "powerful toolset for developing, deploying, and testing Ethereum and EVM-compatible smart contracts". A "Start coding online" button with a right arrow is present.

The main workspace shows a Solidity smart contract named `SimpleStorage.sol`. The code is as follows:

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract SimpleStorage{
5     uint public storedData;
6
7     constructor(uint _data) { Infinite gas 73000 gas
8         storedData = _data;
9     }
10
11     function set(uint x) public { 22514 gas
12         storedData = x;
13     }
14
15     function get() public view returns (uint) { 2493 gas
16         return storedData;
17     }
18 }

```

The interface includes a "FILE EXPLORER" on the left with a tree view showing files like `.deps`, `.states`, `artifacts`, `contracts`, `CryptoSwap`, `scripts`, `tests`, `.prettierrc.json`, `MessageBoard.sol`, `README.txt`, `SimpleStorage.sol` (selected), `simplestorage.sol`, and `TodoList.sol`. The bottom status bar shows "Scam Alert", "Initialize as git repo", "Did you know? To prototype using the Gnosis safe multi sig wallet: create a multisig workspace.", and "RemixAI Copilot (enabled)". The system tray at the bottom indicates the time is 6:04 PM on 8/16/2025.

Implementation Phase: Final Output (no error)

SOLIDITY COMPILER

COMPILER +

0.8.30+commit.73712a01

☐ Include nightly builds

☒ Auto compile

☐ Hide warnings

Advanced Configurations >

Compile SimpleStorage.sol

Compile and Run script

CONTRACT

SimpleStorage (SimpleStorage.sol)

Run Remix Analysis

Run SolidityScan

Publish on IPFS

Publish on Swarm

Compilation Details

ABI Bytecode

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT +

0xF01...aD0cc (0.2469957520...

+ Create Smart Account

GAS LIMIT

☒ Estimated Gas

☐ Custom 3000000

VALUE

0 Wei

CONTRACT

SimpleStorage - SimpleStorage.sol

evm version: prague

Deploy 123

☐ Publish to IPFS

At Address Load contract from Address

Transactions recorded >

Account 1
Sepolia

Deploy a contract

This site wants you to deploy a contract

Estimated changes No changes

Request from remix.ethereum.org

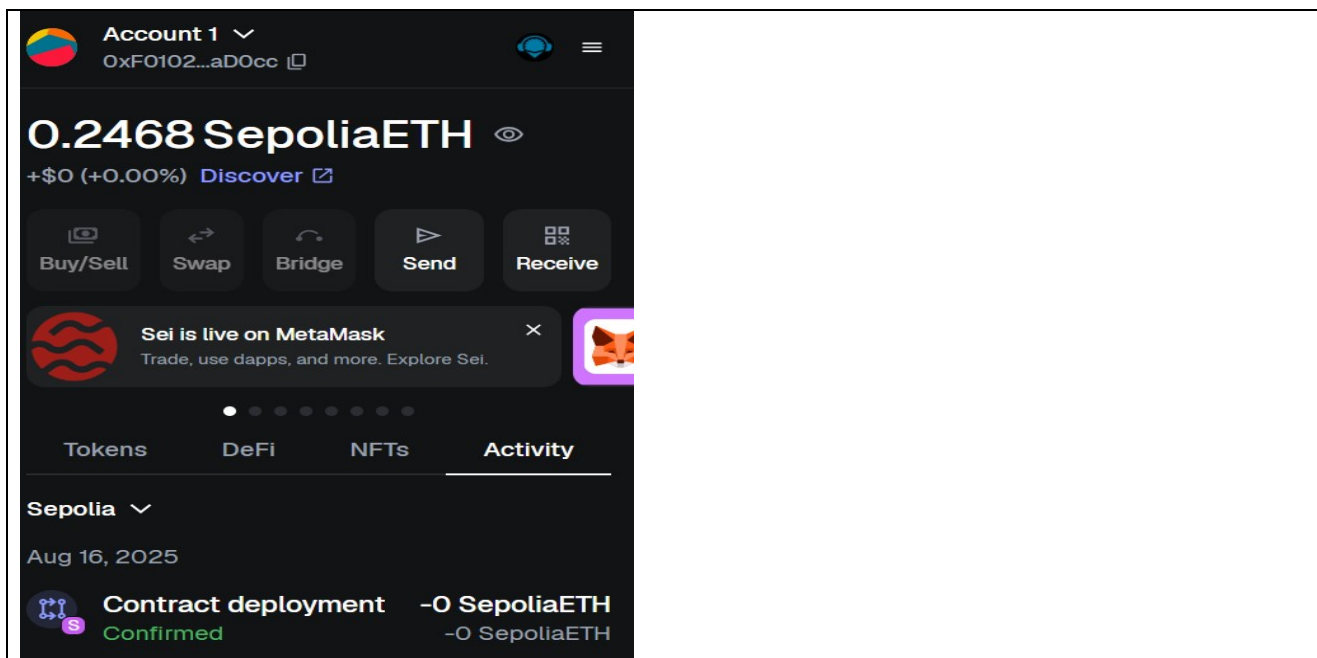
Network fee 0.0002 s SepoliaETH

Speed Market ~16 sec

Cancel Confirm

* Implementation Phase: Final Output (no error)

Applied and Action Learning



* Observations

- After deployment, the smart contract cannot be changed — if there's a bug or error in the code, you must deploy a new version.
- The amount of gas spent varies with how complex the constructor is and how much data is stored or modified.
- Even the simplest contract will leave a permanent record on the blockchain and require gas to execute.
- MetaMask functions not only as a crypto wallet but also as the tool that signs your transactions and sends them to the blockchain network

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.*