



**Centurion**  
UNIVERSITY  
*Shaping Lives... Empowering Communities...*

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

### **Objective/Aim:**

- To write, compile, and deploy your first Solidity smart contract.
- To store and retrieve simple data on the Ethereum Sepolia Testnet using MetaMask and Remix IDE.

### **Apparatus/Software Used:**

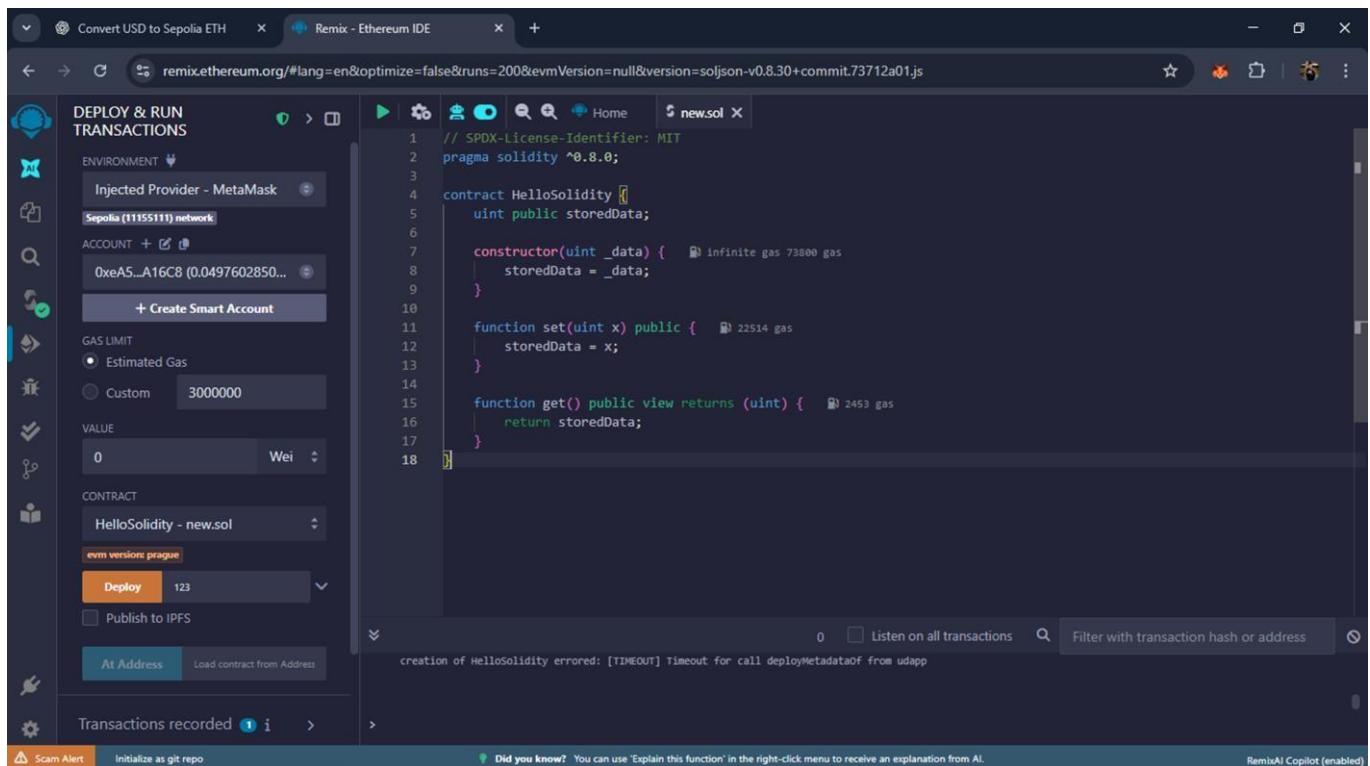
- Laptop/PC
- Word for documentation
- Internet for research
- Chrome Browser
- Remix – Ethereum IDE

### **Theory/Concept:**

- Solidity:** A high-level programming language used to write smart contracts for Ethereum.
- Smart Contract:** A self-executing contract with the rules written directly into code.
- Remix IDE:** A browser-based development environment for writing, deploying, and testing Solidity contracts.
- MetaMask:** A browser extension wallet used to interact with Ethereum-compatible networks like Sepolia.
- Sepolia Testnet:** A public Ethereum test network that simulates the Ethereum mainnet for testing purposes.

## Procedure:

1. **Open Remix IDE** in your browser by visiting <https://remix.ethereum.org>.
2. **Create a new Solidity file** (e.g., new.sol) and write your first smart contract named HelloSolidity.
3. In the contract, define:
  - o A public variable storedData of type uint.
  - o A constructor that accepts a value \_data and stores it in storedData.
  - o A set() function to update the value of storedData.
  - o A get() function to read the current value of storedData.
4. **Connect Remix to MetaMask** by selecting the **Injected Provider - MetaMask** option under the "Environment" dropdown in the Remix sidebar.
5. Make sure **MetaMask is connected to the Sepolia Test Network** and your wallet is unlocked.
6. In Remix, select your contract (HelloSolidity) from the dropdown menu under "CONTRACT".
7. Enter an initial value in the input field next to the **Deploy** button , then click **Deploy** to deploy the contract.



The screenshot shows the Remix Ethereum IDE interface. On the left, the sidebar includes sections for DEPLOY & RUN TRANSACTIONS, CONTRACT (HelloSolidity - new.sol), and various configuration options like GAS LIMIT (Custom, 3000000) and VALUE (0 Wei). The main area displays the Solidity code for the HelloSolidity contract:

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

contract HelloSolidity {
    uint public storedData;

    constructor(uint _data) {
        storedData = _data;
    }

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

Below the code, a message indicates: "creation of HelloSolidity errored: [TIMEOUT] Timeout for call deployMetadataOf from udapp". The bottom status bar shows "Scam Alert" and "Initialize as git repo".

## Observation Table

Observation Point	Details
Remix Environment	Remix IDE (browser-based IDE for Solidity)
Contract Name	HelloSolidity

## 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		
	10		

*Signature of the Faculty:*

*Signature of the Student:*

Name :

Regn.No.