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 : Audit 101 – Smart Contract Vulnerabilities

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Algorithm:

1. Start
2. Open Remix IDE in browser.
3. Create a new Solidity file named VulnerableContract.sol.
4. Write a smart contract with intentional vulnerabilities.
5. Compile the contract and check for warnings or compiler errors.
6. Deploy the vulnerable contract using MetaMask on a test network.
7. Analyze its behavior by performing function calls that exploit the weakness.
8. Identify and record the cause of vulnerability.
9. Modify the contract to fix the issue and redeploy it.

10Re-test the contract to ensure the vulnerability no longer exists.

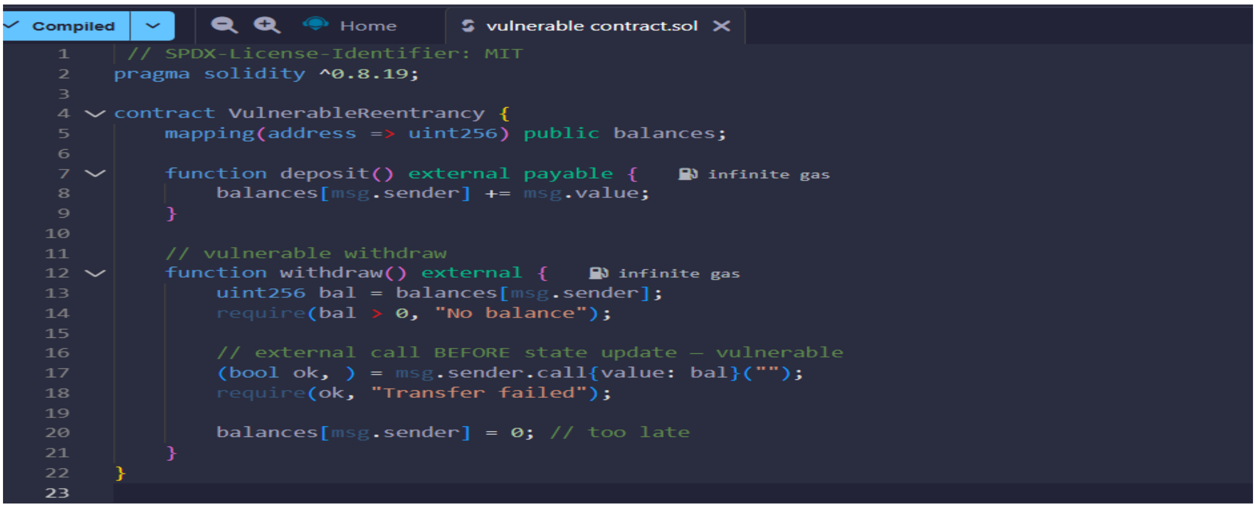
* Software used

1.Remix IDE
2.Solidity
3.MetaMask
4.Test Network (Sepolia)

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

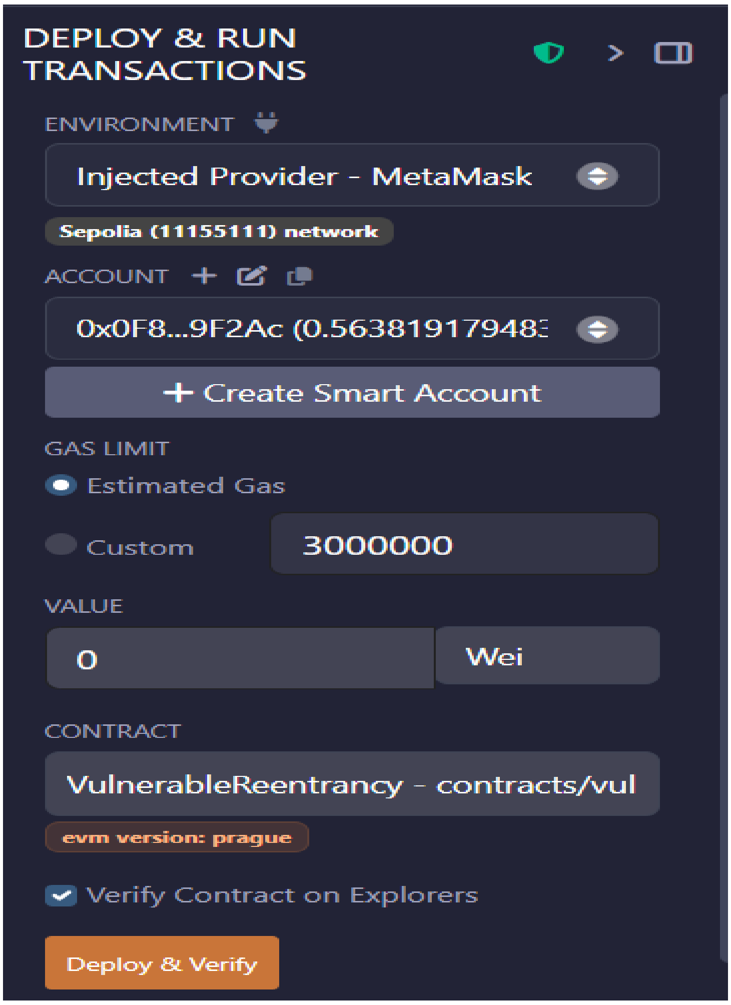
* Testing Phase: Compilation of Code (error detection)

Open Remix IDE. Create a new file VulnerableContract.sol. Write Vulnerable Contract Code



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract VulnerableReentrancy {
    mapping(address => uint256) public balances;

    function deposit() external payable {    infinite gas
        balances[msg.sender] += msg.value;
    }

    // vulnerable withdraw
    function withdraw() external {    infinite gas
        uint256 bal = balances[msg.sender];
        require(bal > 0, "No balance");

        // external call BEFORE state update — vulnerable
        (bool ok, ) = msg.sender.call{value: bal}("");
        require(ok, "Transfer failed");

        balances[msg.sender] = 0; // too late
    }
}
```

Deploy and Test:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0x0F8...9F2Ac (0.563819179483

GAS LIMIT

● Estimated Gas

○ Custom 3000000

VALUE: 0 Wei

CONTRACT: VulnerableReentrancy - contracts/vul

evm version: prague

☑ Verify Contract on Explorers

* Implementation Phase: Final Output (no error)

Successfully analyzed a vulnerable smart contract.
Detected and mitigated a Reentrancy Attack vulnerability.

* Observations

•The smart contract was thoroughly analyzed to identify common vulnerabilities such as reentrancy, integer overflow/underflow, and improper access control.
•Test cases and audit tools detected potential weaknesses that could lead to unauthorized fund access or logical errors.
•Reentrancy checks and input validations were found essential to prevent exploitation.

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 :

Signature of the Faculty:

Regn. No. :

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