**PRACTICAL: 4**

**AIM:** Implement basic of Solidity programming: Syntax, Variables, Functions, mapping, access modifiers using Remix online IDE. Use online REMIX IDE by Ethereum to run solidity code..

**CODE:**

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract BasicConcepts {

// 1. Variables (State Variables)

string public name = "My First Contract"; // public variable

uint private number = 100; // private variable

address public owner; // address variable

bool isActive = true; // boolean variable

// 2. Mapping

mapping(address => uint) public balances;

// 3. Constructor

constructor() {

owner = msg.sender; // msg.sender is the address that deploys the contract

balances[msg.sender] = 1000; // Give initial balance to owner

}

// 4. Access Modifier

modifier onlyOwner() {

require(msg.sender == owner, "Not the owner");

\_;

}

// 5. Functions with different access modifiers

// Public function - anyone can call

function getNumber() public view returns(uint) {

return number;

}

// Private function - only this contract can call

function privateFunction() private pure returns(string memory) {

return "This is private";

}

// Function that only owner can call (using modifier)

function setNumber(uint \_newNumber) public onlyOwner {

number = \_newNumber;

}

// Function to receive balance

function getBalance(address \_address) public view returns(uint) {

return balances[\_address];

}

// Function to transfer balance

function transfer(address \_to, uint \_amount) public {

require(balances[msg.sender] >= \_amount, "Not enough balance");

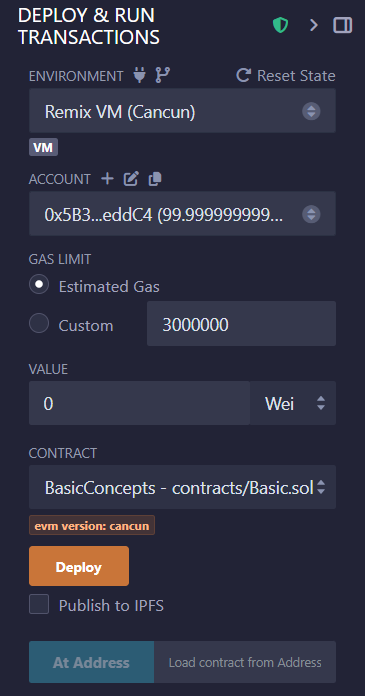
balances[msg.sender] -= \_amount;

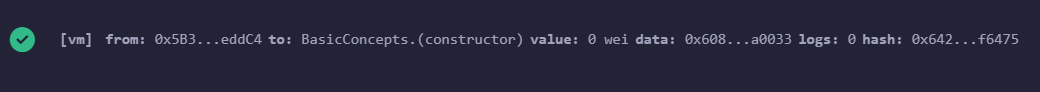
balances[\_to] += \_amount;

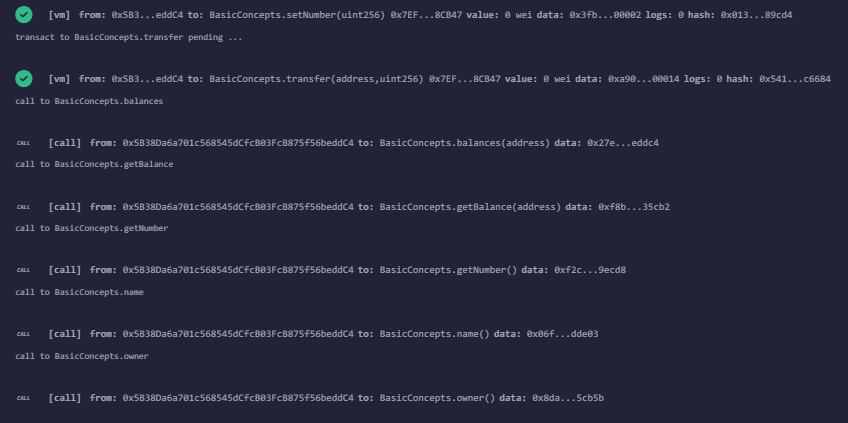
}

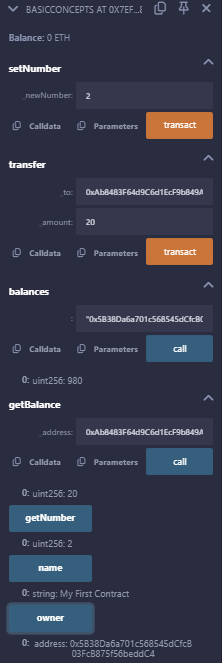
}

**OUTPUT:**

****

****

****



**LEARNING OUTCOME:**

Solidity is the primary programming language for writing smart contracts on the Ethereum blockchain. It follows a contract-oriented approach, where developers define contract structures, variables, functions, and mappings. Key concepts include data types, functions, access modifiers, and mappings that help manage blockchain data securely. Using Remix IDE, developers can write, compile, and test Solidity smart contracts efficiently, enabling hands-on experience in deploying and interacting with blockchain applications.