|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D:\sep2k3\COLLEG~1\LOGO.JPG | SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG. SHEGAON | | | **LABORATORY MANUAL** | |
| **PRACTICAL EXPERIMENT INSTRUCTION SHEET** | | | | |
| LABORATORY MANUAL NO.:SSGMCE/WI/IT/BCF/7IT08/15 | | ISSUE NO.: 00 | | ISSUE DATE: 18/07/2022 | |
| REV.DATE: | | DEPTT:INFORMATION TECHNOLOGY | | | |
| LABORATORY: Block Chain Fundamental | | | SEMESTER: 4N/VII | | PAGE: 1 OF 3 |

**AIM:** Compilation & deployment of Smart Contract in Solidity using Remix IDE

Requirement: Remix IDE

1. **THEORY**

Solidity is a contract-oriented, high-level programming language for implementing smart contracts. Solidity is highly influenced by C++, Python and JavaScript and has been designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types programming language. You can use Solidity to create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

## What is Ethereum?

Ethereum is a decentralized ie. blockchain platform that runs smart contracts i.e. applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference.

## The Ethereum Virtual Machine (EVM)

The Ethereum Virtual Machine, also known as EVM, is the runtime environment for smart contracts in Ethereum. The Ethereum Virtual Machine focuses on providing security and executing untrusted code by computers all over the world.

The EVM specialised in preventing Denial-of-service attacks and ensures that programs do not have access to each other's state, ensuring communication can be established without any potential interference.

The Ethereum Virtual Machine has been designed to serve as a runtime environment for smart contracts based on Ethereum.

## What is Smart Contract?

A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible. The concept of smart contracts was first proposed by Nick Szabo in 1994. Szabo is a legal scholar and cryptographer known for laying the groundwork for digital currency. A Solidity source files can contain an any number of contract definitions, import directives and pragma directives.

pragma solidity >=0.4.0 <0.6.0;

contract SimpleStorage {

uint storedData;

function set(uint x) public {

storedData = x;

}

function get() public view returns (uint) {

return storedData;

}

}

## Pragma

The first line is a pragma directive which tells that the source code is written for Solidity version 0.4.0 or anything newer that does not break functionality up to, but not including, version 0.6.0.

A pragma directive is always local to a source file and if you import another file, the pragma from that file will not automatically apply to the importing file.

## Contract

A Solidity contract is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereumblockchain.

The line uintstoredData declares a state variable called storedData of type uint and the functions set and get can be used to modify or retrieve the value of the variable.

## Importing Files

Though above example does not have an import statement but Solidity supports import statements that are very similar to those available in JavaScript.

The following statement imports all global symbols from "filename".

1. **CONCLUSION:**. In this way,we studies Compilation & deployment of Smart Contract in Solidity using Remix IDE.

pragma solidity ^0.5.0;

contract SolidityTest {

constructor() public{

}

function getResult() public pure returns(uint){

uint a = 1;

uint b = 2;

uint result = a + b;

return result;

}

}