COMP BEB1326
72017912C

GROUP C : ASSIGNMENT No. 4

Page :
Date : / /

Title: Solidity Program

Objective: To understand how to write a program in solidity & dploy it as smart contract on Ethereum.

Problem Statement:

Write a program in solidity to create Student data. Use the following constructs:

1> Structures
2> Arrays
3> Fallback

Deploy this as smart contract on Ethereum & observe the transcaction Fee & gas values.

Software & Hardware Requirement:

1> Desktop/Laptop
2> Any Operating System
3> Internet Connection
4> IDE

Theory:

Solidity:

Solidity is a brand-new programming language created by the Ethereum which is the second-largest market of cryptocurrency by capitalization, released in the year 2015 led by Christian Reitwiesner. Some key features of solidity are listed below:

1> Solidity is high-level programming language designed for implementing smart contracts.

2> It is statically-typed object-oriented (contract-oriented) language.

3> Solidity is highly influenced by Python, C++, & javascript which runs on the Ethereum Virtual Machine (EVM).

4> Solidity supports complex-user-defined programming, libraries & inheritance.

5> Solidity is primary language for blockchains running platforms.

6> Solidity can be used to creating contracts like voting, blind auctions, crowdfunding, multi-signature wallets, etc.

Ethereum:

Ethereum is a decentralized open-source platform based on blockchain domain, used to run smart contracts i.e. applications that execute the program exactly as it was programmed without the possibility of any fraud, interference from a third party, censorship, or downtime. It serves a platform for nearly 2,60,000 different cryptocurrencies. Ether is a cryptocurrency generated by ethereum miners, used to reward for the computations performed to secure the blockchain.

Ethereum Virtual Machine (EVM):

Ethereum virtual machine abbreviated as EVM is a runtime environment for executing smart contracts in ethereum. It focuses widely on providing security & execution of untrusted code using an international network of public nodes. EVM is specialized to prevent Denial-of-service attack & confirms that the program does not have any access to each other's state, also ensures that the communication is established without any potential interference.

Smart Contract:

Smart contracts are high-level program codes that are compiled to EVM byte code & deployed to the ethereum blockchain for further execution. It allows us to perform credible transactions without any interference of the third party, these transactions are trackable & irreversible. Languages used to write smart contracts are Solidity (a language library with similarities to C & Javascript), Serpent (Similar to Python, but deprecated), LLL (a low-level lisp-like language), & Mutan (Go-based, but deprecated).

Smart Contract Working steps:

1> Identify agreement
2> Set conditions
3> Code business logic
4> Encryption & blockchain technology
5> Execution & processing
6> Network updates.

Applications of Smart Contracts:

1> Real-Estate
2> Vehicle Ownership
3> Music Industry
4> Government Elections
5> Management
6> Healthcare

Advantages of Smart Contracts:

1> Recordkeeping
2> Autonomy
3> Reduce Fraud
4> Fault-tolerance
5> Enhanced trust
6> Cost-efficiency.

Conclusion:

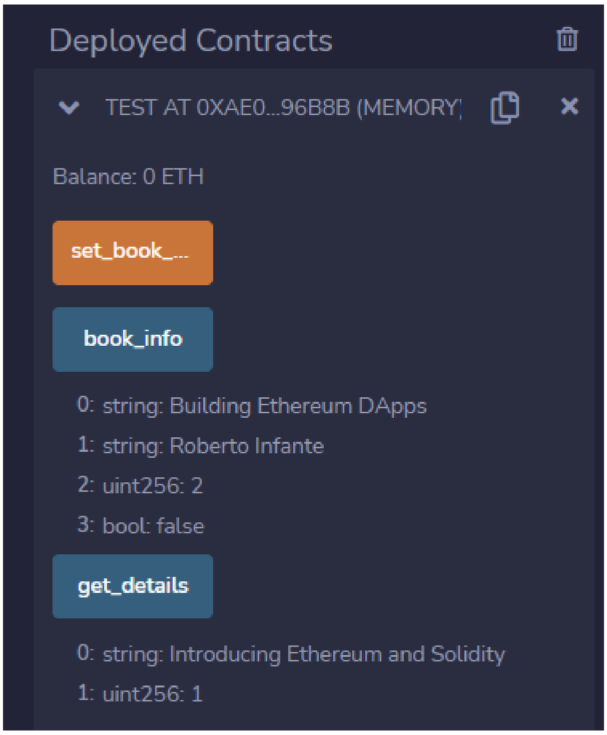
Successfully wrote a program in solidity & deployed it as smart contract on Etheream.

17/10/22

Code:

```
// Solidity program to demonstrate
// how to use 'structures'
pragma solidity ^0.5.0;
// Creating a contract
contract test {
    // Declaring a structure
    struct Book {
        string name;
        string writter;
        uint256 id;
        bool available;
    }

    // Declaring a structure object
    Book book1;

    // Assigning values to the fields
    // for the structure object book2
    Book book2 = Book("Building Ethereum DApps", "Roberto Infante ", 2, false);

    // Defining a function to set values
    // for the fields for structure book1
    function set_book_detail() public {
        book1 = Book(
            "Introducing Ethereum and Solidity",
            "Chris Dannen",
            1,
            true
        );
    }

    // Defining function to print
    // book2 details
    function book_info()
        public
        view
        returns (
            string memory,
            string memory,
            uint256,
            bool
        )
    {
        return (book2.name, book2.writter, book2.id, book2.available);
    }

    // Defining function to print
    // book1 details
    function get_details() public view returns (string memory, uint256) {
        return (book1.name, book1.id);
    }
}
```

Output:



```
Balance: 0 ETH
0: string: Building Ethereum DApps
1: string: Roberto Infante
2: uint256: 2
3: bool: false
0: string: Introducing Ethereum and Solidity
1: uint256: 1
```