Practical No 3.

AIM: Write a smart contract on a test network, for Bank account of a customer for following operations:

• Deposit money

• Withdraw Money

• Show balance

Objective: Understand and explore the working of Blockchain Technology and its Application

Course Outcome:

CO6: Interpret the basic concepts in Blockchain Technology and its Application.

Theory:

Smart Contract?

A Smart Contract (or cryptocontract) is a computer program that directly and automatically controls the transfer of digital assets between the parties under certain conditions.

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.

Smart Contract Working

- **Identify Agreement:** Multiple parties identify the cooperative opportunity and desired outcomes and agreements could include business processes, asset swaps, etc.
- **Set conditions:** Smart contracts could be initiated by parties themselves or when certain conditions are met like financial market indices, events like GPS locations, etc.
- **Code business logic:** A computer program is written that will be executed automatically when the conditional parameters are met.
- **Encryption and blockchain technology:** Encryption provides secure authentication and transfer of messages between parties relating to smart contracts.
- **Execution and processing:** In blockchain iteration, whenever consensus is reached between the parties regarding authentication and verification then the code is executed and the outcomes are memorialized for compliance and verification.

- **Network updates:** After smart contracts are executed, all the nodes on the network update their ledger to reflect the new state. Once the record is posted and verified on the blockchain network, it cannot be modified, it is in append mode only.

Smart Wallet:

Wallets are just like your bank account, through which we can receive money, send money, and can check the balance.

Solidity Programming Concepts:

Solidity's code is encapsulated in contracts which means a contract in Solidity is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. A contract is a fundamental block of building an application on Ethereum.

1. **Spdx license identifier**

SPDX License Identifiers can be **used to indicate relevant license information at any level, from package to the source code file level**.

2. **Version Pragma:**

pragma solidity >=0.4.16 <0.7.0;

Pragmas are instructions to the compiler on how to treat the code. All solidity source code should start with a "version pragma" which is a declaration of the version of the solidity compiler this code should use.

This helps the code from being incompatible with the future versions of the compiler which may bring changes. The above-mentioned code states that it is compatible with compilers of version greater than and equal to 0.4.16 but less than version 0.7.0.

3. **The contract keyword:**

contract Storage{

//Functions and Data

}

The contract keyword declares a contract under which is the code encapsulated.

4. **State variables:**

uint public setData;

State variables are permanently stored in contract storage that is they are written in Ethereum Blockchain.

The line uint setData declares a state variable called setData of type uint (unsigned integer of 256 bits). Think of it as adding a slot in a database.

5. **A function declaration:**

function set(uint x) public

function get() public view returns (uint)

This is a function named *set* of access modifier type *public* which takes a variable *x* of datatype *uint* as a parameter. This was an example of a simple smart contract which updates the value of setData. Anyone can call the function set and overwrite the value of setData which is stored in Ethereum blockchain and there is possibly no way for anyone to stop someone from using this function.Function get will retrieve and print the value of the state variable.

Important terms:

1. **uint-** unsigned integer.
2. **address-** It is a unique set of the string provided for the transaction.
3. **msg.sender-** Address of the current user.
4. **msg.value-** Cost put by the current user.
5. **public-** Visibility of the function.
6. **view-** This keyword is used when the function is read-only.
7. **modifier-** modifier is just like a function, which is used at, the initial point of function so that it can check the required condition and prevent wastage of ether before function execution.

Conclusion:

Printout of Program code

Output screen shot paste below