

Practical 04

Problem Statement:

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

- Structure
- Arrays
- Fallback

Objective:

Understand and explore the working of Blockchain technology and its applications.

Course Outcome:

CO6: Interpret the basic concepts in Blockchain technology and its applications.

Description:

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

A smart contract is just a digital contract with the security coding of the blockchain.

- It has details and permissions written in code that require an exact sequence of events to take place to trigger the agreement of the terms mentioned in the smart contract.
- It can also include the time constraints that can introduce deadlines in the contract.
- Every smart contract has its address in the blockchain. The contract can be interacted with by using its address presuming the contract has been broadcasted on the network.

The idea behind smart contracts is pretty simple. They are executed on a basis of simple logic, IF-THEN for example:

- **IF** you send object A, **THEN** the sum (of money, in cryptocurrency) will be transferred to you.
- **IF** you transfer a certain amount of digital assets (cryptocurrency, for example, ether, bitcoin), **THEN** the A object will be transferred to you.
- **IF** I finish the work, **THEN** the digital assets mentioned in the contract will be transferred to me.

Code:

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.6;
contract Student_management{
    struct Student{
        int stud_id;
        string Name;
        string department;
    }

    Student[] Students;

    function add_stud(int stud_id, string memory Name, string memory department) public{
        Student memory stud = Student(stud_id, Name, department);
        Students.push(stud);
    }

    function getStudent(int stud_id) public view returns(string memory, string memory){
        for(uint i=0; i< Students.length; i++){
            Student memory stud = Students[i];
            if(stud.stud_id == stud_id){
                return(stud.Name, stud.department);
            }
        }
        return("Student Information not found...!", "Not Found");
    }
}
```

OUTPUT:

```
[vm] from: 0x583...eddC4 to: Student_management.(constructor) value: 0 wei data: 0x608...c0033 logs: 0
hash: 0xfa5...f21d9

status          true Transaction mined and execution succeed
transaction hash 0xfa591e58975583f54477867bfdc5c9233d95be71d11b7fbc1661d88d3f3f21d9
from            0x5838Da6a701c568545dCfcB03FcB875f56beddC4
to              Student_management.(constructor)
gas             471422 gas
transaction cost 409932 gas
execution cost  409932 gas
input           0x608...c0033
decoded input    {}
decoded output   -
```

```
✓ [vm] from: 0x583...eddC4 to: Student_management.add_stud(int256,string,string) 0xD4F...2cbee value: 0 wei
data: 0xbfc...00000 logs: 0 hash: 0xcb0...6b4a7

status      true Transaction mined and execution succeed

transaction hash  0xcb088c06147b1eb68dfe4bd7db7985453d9c491ca15b6a784007f96ad2f6b4a7 ⓘ

from          0x5838Da6a701c568545dCfcB03FcB875f56beddC4 ⓘ

to            Student_management.add_stud(int256,string,string) 0xD4Fc541236927E2EAF8F27606bD7309C1Fc2cbee ⓘ

gas           109894 gas ⓘ

transaction cost  95560 gas ⓘ

execution cost   95560 gas ⓘ

input          0xbfc...00000 ⓘ

decoded input    {
                  "int256 stud_id": "1230",
                  "string Name": "John",
                  "string department": "Computer Engineering"
                } ⓘ

decoded output   {} ⓘ

call to Student_management.getStudent

CALL [call] from: 0x5838Da6a701c568545dCfcB03FcB875f56beddC4 to: Student_management.getStudent(int256) data: 0xce5...004ce

from          0x5838Da6a701c568545dCfcB03FcB875f56beddC4 ⓘ

to            Student_management.getStudent(int256) 0xD4Fc541236927E2EAF8F27606bD7309C1Fc2cbee ⓘ

execution cost 39950 gas (Cost only applies when called by a contract) ⓘ

input          0xce5...004ce ⓘ

decoded input    {
                  "int256 stud_id": "1230"
                } ⓘ

decoded output   {
                  "0": "string: John",
                  "1": "string: Computer Engineering"
                } ⓘ

logs            [] ⓘ ⓘ
```

Conclusion:

I studied about smart contract and concept such as array, fallback and how to write and execute it using remix ide.