

ASSIGNMENT – 4

(Blockchain Technology)



Submitted to: Mr. Shashikant

Name: Ishit Singh

Class: 3NC1

Roll No.: 102115023

Semester: Jan'24-May'24

CODE:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract ERC20{

    address contractRunner;           // address of contract/token owner
    string public name;                // name of the token
    string public symbol;              // symbol of the token
    uint8 public immutable decimals;  // number of decimal places that the
token can be divided into [IMMUTABLE ---> Can't be altered again]
    uint public immutable totalSupply; // total supply of tokens that have
been created for a particular project

    /* Mapping is a hash table in Solidity that stores data as key-value
pairs, where the key can be any of the built-in data types
    1. Never let an array in Solidity grow too large because iterating
through a large array could cost more in Solidity gas fees than the
transaction's value,
    2. making mappings a more gas efficient smart contract implementation
    */
    mapping(address => uint) balances;           // Account
address --- has count of tokens ---> uint
    // spender => (owner => no. of tokens allowed)
    mapping(address => mapping(address => uint)) allowances; // Account
address --- has delegated account --> address --- to have tokens amount of ---
> uint

    // EVENTS : Data stored on blockchain that is only to be STORED and
neither Modified OR Accessed
    event Transfer(address indexed _from, address indexed _to, uint value);
    event Approval(address indexed _owner, address indexed _spender, uint
value);

    constructor(string memory _name, string memory _symbol, uint8 _decimals,
uint _totalSupply){
        contractRunner = msg.sender;
        name = _name;
        symbol = _symbol;
        decimals = _decimals;
        totalSupply = _totalSupply;
        balances[contractRunner] = totalSupply;
    }

    // function returns the balance of tokens held by a particular address.
    function balanceOf(address _owner) public view returns(uint){
        require(_owner != address(0), "Zero Address");
        return balances[_owner];
    }
}
```

```

    }

    // function allows an address to send tokens to another address.
    function transfer(address _to, uint _tokenVal) public returns(bool){
        require((balances[contractRunner] >= _tokenVal) &&
(balances[contractRunner] >= 0), "Insufficient Balance");

        balances[contractRunner] -= _tokenVal;
        balances[_to] += _tokenVal;
        emit Transfer(contractRunner, _to, _tokenVal);
        return true;
    }

    // function allows an address [that has approved of transfer] to transfer
tokens from one address to another address.
    function transferFrom(address _from, address _to, uint tokenVal) public
returns(bool){
        require(allowances[contractRunner][_from] >= tokenVal, "Tokens
transfer not allowed");
        require((balances[_from] >= tokenVal) && (balances[_from] > 0),
"Insufficient Balance");

        balances[_from] -= tokenVal;
        balances[_to] += tokenVal;
        allowances[contractRunner][_from] -= tokenVal;
        emit Transfer(_from, _to, tokenVal);
        return true;
    }

    // function allows an address(msg.sender) to approve another
address(spender) to spend tokens on their behalf.
    function approve(address _spender, uint _tokenVal) public returns(bool){
        require(balances[contractRunner] >= _tokenVal, "Insufficient balance");
        allowances[_spender][contractRunner] = _tokenVal;
        emit Approval(contractRunner, _spender, _tokenVal);
        return true;
    }

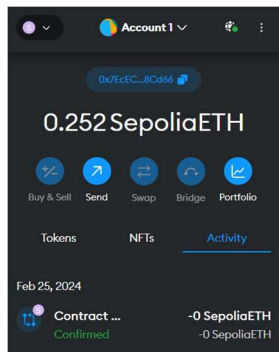
    // function returns the amount of tokens that an approved address can
spend on behalf of another address.
    function allowanceFunc(address _owner, address _spender) public view
returns(uint){
        return allowances[_spender][_owner];
    }
}

```

OUTPUT:

1. Connecting with METAMASK

<div> <div> Latest 4 from a total of 4 transactions </div> <div> Download Page Data </div> </div>							
Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0x0beaf7990d7987a22...	0x60c0c06040	5361437	18 secs ago	0x7EcEc3...b338Cd66	OUT Contract Creation	0 ETH	0.0036256



2. Deploying Contract

```
[block:5361437 txIndex:85] from: 0x7ec...8cd66 to: ERC20.(constructor) value: 0 wei data:
```

status	0x1 Transaction mined and execution succeed
transaction hash	0xb0eaf799bd7987a2240f0504968efa736d0250583ea8453bf5af05df86ba34e ⓘ
block hash	0x31ebc769349b44cb298B0a7e5ee3dd3ed10676E8afcc04410d3d60fcf8e23771 ⓘ
block number	5361437 ⓘ
contract address	0xe5fa8f789374954197878cbf72d135db1a756d1 ⓘ
from	0x7ecce3ac6460706367663e64385ce8751b3cd8d6 ⓘ
to	ERC20.(constructor) ⓘ
gas	1157263 gas ⓘ
transaction cost	1146993 gas ⓘ
input	0x60c...00000 ⓘ
decoded input	{ "string_name": "LEO", "string_symbol": "L", "uint8_decimals": 18, "uint256_totalSupply": "1000000000000000000000" } ⓘ
decoded output	- ⓘ
logs	[] ⓘ ⓘ

3. Running Functions

[illegible]

```
[block:5361522 txIndex:52] from: 0x7ec...8cd66 to: ERC20.transfer(address,uint256) 0xe5f...756d1
hash: 0x899...3f873

status                                0x1 Transaction mined and execution succeed

transaction hash                       0x62e51962efad70c75f3b51cdf23c18126f373f33a3349547ddbdf9cc62c1a78 ⓘ

block hash                            0x8998ae3f8e7fe5e4f538998ab883a4ad6ec7f4f5844749a5e970ad6cfa3f873 ⓘ

block number                          5361522 ⓘ

from                                  0x7ec9c36c460706367663e64385ce8751b338cd66 ⓘ

to                                    ERC20.transfer(address,uint256) 0xe5fa8f78934954197878cbf72d135d1b1a756d1 ⓘ

gas                                    55912 gas ⓘ

transaction cost                       55110 gas ⓘ

input                                  0xa90...003e8 ⓘ

decoded input                          {
                                         "address" to: "0xd1bd27c9bE2943e8ec0ce43d6F88f99c434EEb7",
                                         "uint256 _tokenVal": "1000"
                                     } ⓘ

decoded output                         - ⓘ

logs                                   [
```

[This is a Sepolia **Testnet** transaction only]

Transaction Hash:	0x62e51962ef0d70c75f3b51cdf23c18126f373f33a3349547ddbdef9cc62c1a78
Status:	Success
Block:	5361522 6 Block Confirmations
Timestamp:	1 min ago (Feb-25-2024 05:02:48 PM +UTC)
Transaction Action:	Call Transfer Function by 0x7EcEC3...b338Cd66 on 0xE5FA8f...b1A756D1
From:	0x7EcEC36c460706367663e64385cE8751b338Cd66
Interacted With (To):	0xE5FA8f789374954197878Cbf72d135D1b1A756D1
ERC-20 Tokens Transferred:	<div>All Transfers Net Transfers</div> <p>0x7EcEC3...b338Cd66 sent 0.0000000000000001 LEO... (L...)</p> <p>0xd1bd27...e434EEb7 received 0.0000000000000001 LEO... (L...)</p>
Value:	0 ETH (\$0.00)
Transaction Fee:	0.00017091174514788 ETH (\$0.00)
Gas Price:	3.101283708 Gwei (0.000000003101283708 ETH)

Fig a) Sending 1000 “LEO” tokens [Transfer]

[This is a Sepolia **Testnet** transaction only]

Transaction Hash:	0x5c25ce77cfb96778df69329799fcc65f9e5a7070240d2ba3eb95795b28da6a6d
Status:	Success
Block:	5361557 4 Block Confirmations
Timestamp:	54 secs ago (Feb-25-2024 05:10:12 PM +UTC)
Transaction Action:	Call Approve Function by 0x7EcEC3...b338Cd66 on 0xE5FA8f...b1A756D1
From:	0x7EcEC36c460706367663e64385cE8751b338Cd66
To:	0xE5FA8f789374954197878Cbf72d135D1b1A756D1
Value:	0 ETH (\$0.00)
Transaction Fee:	0.0001849174049415 ETH (\$0.00)
Gas Price:	3.6035741 Gwei (0.0000000036035741 ETH)

```
CALL [call] from: 0x7EcEC36c460706367663e64385cE8751b338Cd66 to: ERC20.allowanceFunc(address,address)

from      0x7EcEC36c460706367663e64385cE8751b338Cd66

to      ERC20.allowanceFunc(address,address) 0xE5FA8f789374954197878Cbf72d135D1b1A756D1

input      0x087...4eeb7

decoded input
{
  "address_owner": "0x7EcEC36c460706367663e64385cE8751b338Cd66",
  "address_spender": "0xd1bd27c9bE2943e8ec0ce43d6F8B8f9Ce434EEb7"
}

decoded output
{
  "0": "uint256: 1000000000"
}

logs      []
```

Fig b) Giving allowance to another account [Approve] & displaying the allowance value [AllowanceFunc]