Centurion UNIVERSITY
Shaping Lives... Empowering Communities...

School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ERC-20 Token Standard

Tokens can represent virtually anything in Ethereum:

- reputation points in an online platform
- skills of a character in a game
- financial assets like a share in a company
- a fiat currency like USD
- an ounce of gold
- and more...

Such a powerful feature of Ethereum is handled by a robust standard, ERC-20 plays this role. This standard allows developers to build token applications that are interoperable with other products and services. The ERC-20 standard is also used to provide additional functionality to ether.

ERC-20:

- ➢ The ERC-20 introduces a standard for Fungible Tokens, in other words, they have a property that makes each Token be exactly the same (in type and value) as another Token.
- ➢ For example, an ERC-20 Token acts just like the ETH, meaning that 1 Token is and will always be equal to all the other Tokens.

Prerequisites

- Accounts
- Smart Contracts
- Token standards

❖ The ERC-20 (Ethereum Request for Comments 20), proposed by Fabian Vogelsteller in November 2015, is a Token Standard that implements an API for tokens within Smart Contracts.

Example functionalities ERC-20 provides:

- transfer tokens from one account to another
- get the current token balance of an account
- get the total supply of the token available on the network
- approve whether an amount of token from an account can be spent by a third-party account

If a Smart Contract implements the following methods and events it can be called an ERC-20 Token Contract and, once deployed, it will be responsible to keep track of the created tokens on Ethereum.

Coding Phase: Pseudo Code / Flow Chart / Algorithm

Methods

```
        function name() public view returns (string)
        function symbol() public view returns (string)
        function decimals() public view returns (uint8)
        function totalSupply() public view returns (uint256)
        function balanceOf(address _owner) public view returns (uint256 balance)
        function transfer(address _to, uint256 _value) public returns (bool success)
        function transferFrom(address _from, address _to, uint256 _value) public returns (bool
success)
        function approve(address _spender, uint256 _value) public returns (bool success)
        function allowance(address _owner, address _spender) public view returns (uint256
remaining)
```

Events

```
       event Transfer(address indexed _from, address indexed _to, uint256 _value)
       event Approval(address indexed _owner, address indexed _spender, uint256 _value)
```

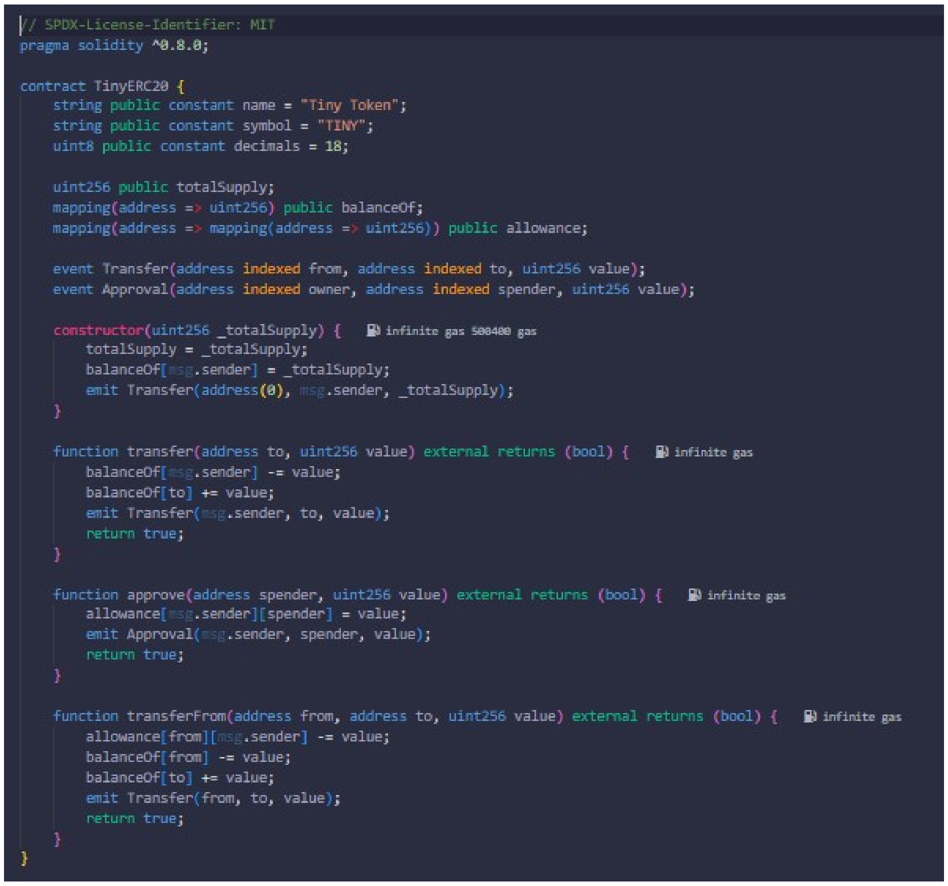
* Softwares used

- Solidity
- Solc compiler
- JSON
- ERC20 Token Standard
- Remix IDE
- Brave Browser
- Metamask
-

*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

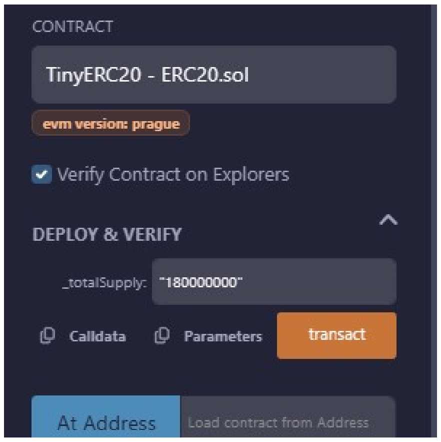
* Implementation Phase: Final Output (no error)

This contract implements only the absolute minimum required functions and events to be a valid, functional ERC-20 token :



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract TinyERC20 {
    string public constant name = "Tiny Token";
    string public constant symbol = "TINY";
    uint8 public constant decimals = 18;

    uint256 public totalSupply;
    mapping(address => uint256) public balanceOf;
    mapping(address => mapping(address => uint256)) public allowance;

    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(address indexed owner, address indexed spender, uint256 value);

    constructor(uint256 _totalSupply) {    infinite gas 500400 gas
        totalSupply = _totalSupply;
        balanceOf[msg.sender] = _totalSupply;
        emit Transfer(address(0), msg.sender, _totalSupply);
    }

    function transfer(address to, uint256 value) external returns (bool) {    infinite gas
        balanceOf[msg.sender] -= value;
        balanceOf[to] += value;
        emit Transfer(msg.sender, to, value);
        return true;
    }

    function approve(address spender, uint256 value) external returns (bool) {    infinite gas
        allowance[msg.sender][spender] = value;
        emit Approval(msg.sender, spender, value);
        return true;
    }

    function transferFrom(address from, address to, uint256 value) external returns (bool) {    infinite gas
        allowance[from][msg.sender] -= value;
        balanceOf[from] -= value;
        balanceOf[to] += value;
        emit Transfer(from, to, value);
        return true;
    }
}
```

Deployed this contract with _totalSupply = 1000000 * 10**18, any wallet or exchange could immediately interface with it using the standard ERC-20 ABI, simply by knowing its contract address.



CONTRACT

TinyERC20 - ERC20.sol

evm version: prague

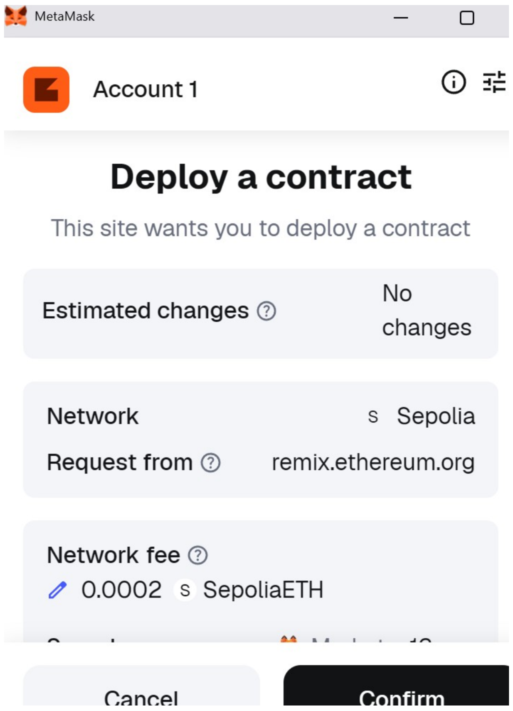
✓ Verify Contract on Explorers

DEPLOY & VERIFY

_totalSupply: "180000000"

*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

Implementation Phase: Final Output (no error)



MetaMask

Account 1

Deploy a contract

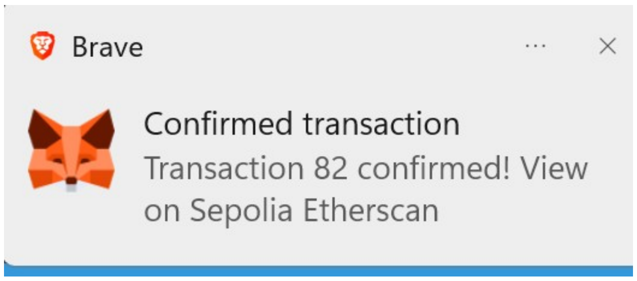
This site wants you to deploy a contract

Estimated changes: No changes

Network: Sepolia

Request from: remix.ethereum.org

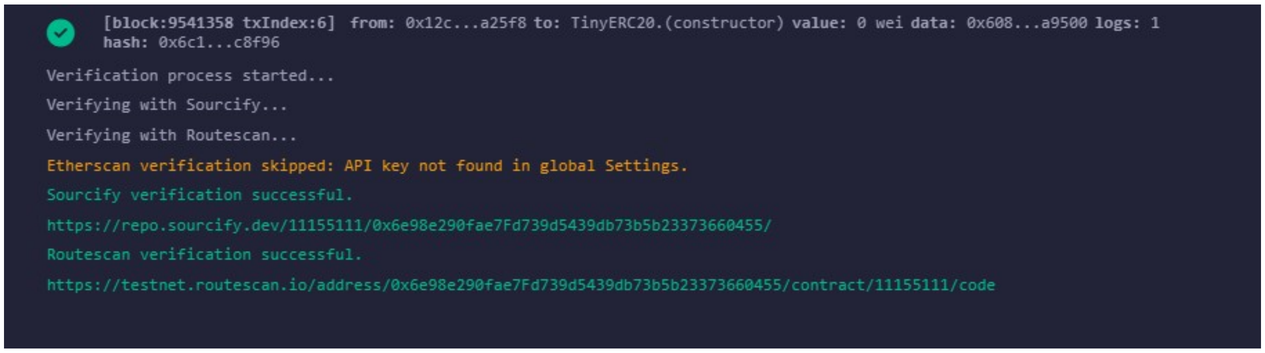
Network fee: 0.0002 SepoliaETH



Brave

Confirmed transaction

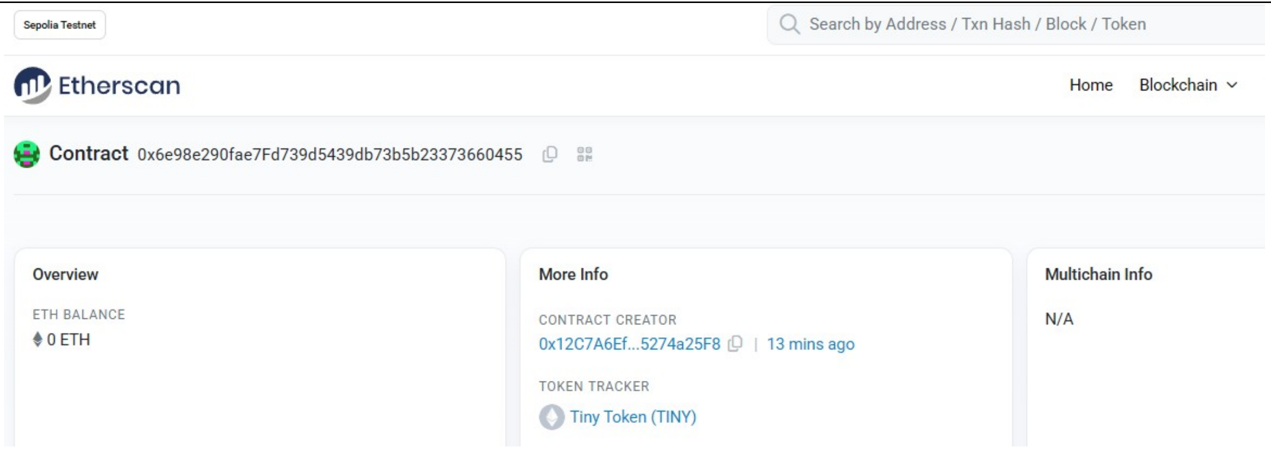
Transaction 82 confirmed! View on Sepolia Etherscan



```
[block:9541358 txIndex:6]  from: 0x12c...a25f8 to: TinyERC20.(constructor) value: 0 wei data: 0x608...a9500 logs: 1
hash: 0x6c1...c8f96
Verification process started...
Verifying with Sourcify...
Verifying with Routescan...
Etherscan verification skipped: API key not found in global Settings.
Sourcify verification successful.
https://repo.sourcify.dev/11155111/0x6e98e290fae7Fd739d5439db73b5b23373660455/
Routescan verification successful.
https://testnet.routescan.io/address/0x6e98e290fae7Fd739d5439db73b5b23373660455/contract/11155111/code
```

*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)


Sepolia Testnet

Search by Address / Txn Hash / Block / Token

Etherscan

Home Blockchain

Contract 0x6e98e290fae7Fd739d5439db73b5b23373660455

Overview

ETH BALANCE
0 ETH

More Info

CONTRACT CREATOR
0x12C7A6Ef...5274a25F8 | 13 mins ago

TOKEN TRACKER
Tiny Token (TINY)

Multichain Info

N/A

* Observations

- Fixed total supply (standard practice for many tokens)
- 18 decimal places (most common ERC-20 choice)
- MetaMask (balance display & transfers)
- Uniswap (trading & liquidity)
- Etherscan (token tracking)
- All DEXs and CEXs supporting ERC-20
- All wallets (Trust Wallet, Coinbase Wallet, etc.)

ASSESMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Regn. No. :

Signature of the Faculty:

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.