

ERC-20 Basics – Tokenization Concepts

School: ............................................................................................................. Campus: ....................................................... Academic Year: ...................... Subject Name: ........................................................... Subject Code: ..........................

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

(Learning by Doing and Discovery)

**\* Coding Phase: Pseudo Code / Flow Chart / Algorithm**

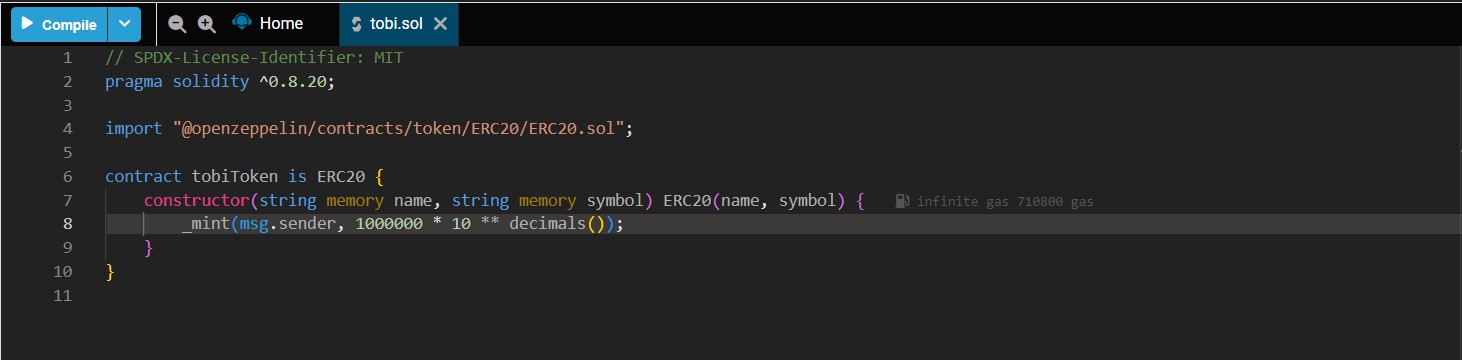
ALGORITHM:

1. **Initialize Development Environment**
   * Launch the Remix IDE in the browser.
2. **Create Smart Contract File**
   * Make a new Solidity file and set up the ERC-20 token code using OpenZeppelin.
3. **Compile the Smart Contract**
   * Use the Solidity compiler (version 0.8.20) and resolve any errors.
4. **Connect Wallet**
   * Choose **Injected Provider – MetaMask** as the environment and connect the wallet.
5. **Deploy Token Contract**
   * Deploy the contract by providing token details (name, symbol) and approve the MetaMask transaction.
6. **Record Deployed Address**
   * Copy the contract address generated after deployment.
7. **Verify on Blockchain Explorer**
   * Paste the contract address in Etherscan/Testnet Explorer to confirm deployment.
8. **Add Token to Wallet**
   * In MetaMask, import the deployed contract address to display the token balance.
9. **Perform Transactions**
   * Use the transfer function from Remix to send tokens to another wallet.
10. **Confirm Transaction Success**
11. Verify the token transfer on Etherscan to ensure proper execution.

# \* Software Used:

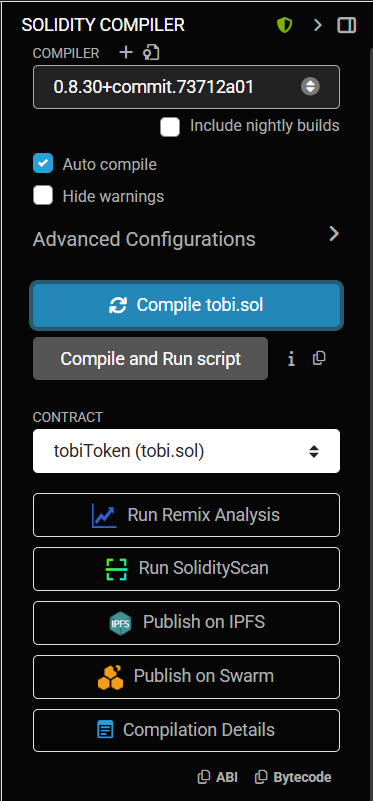
1. **Remix IDE** – For writing, compiling, and deploying the contract.
2. **MetaMask Wallet** – For interacting with Ethereum blockchain.
3. **Etherscan** – For verifying deployed contract and transactions.
4. **OpenZeppelin Contracts** – For ERC-20 standard implementation.
5. **Brave Browser** – To access Remix, MetaMask, and Etherscan.

1. **Open Remix IDE (**[**https://remix.ethereum.org**](https://remix.ethereum.org)**).**
2. In the left panel, click Contracts to create a new Solidity file (e.g. tobi.sol).
3. Write the ERC-20 token code (using OpenZeppelin standard).



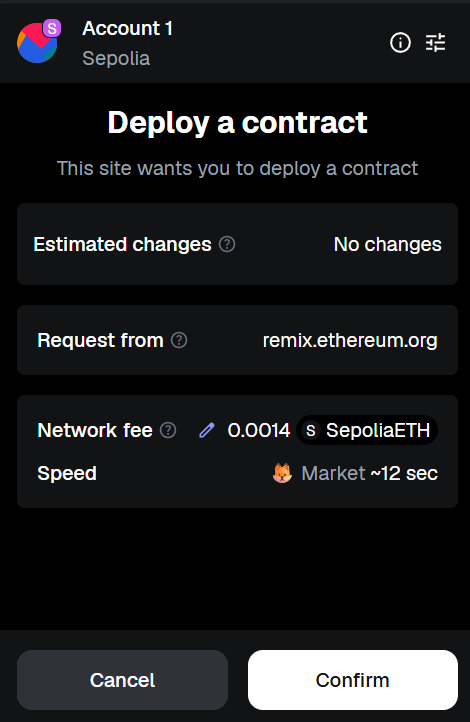
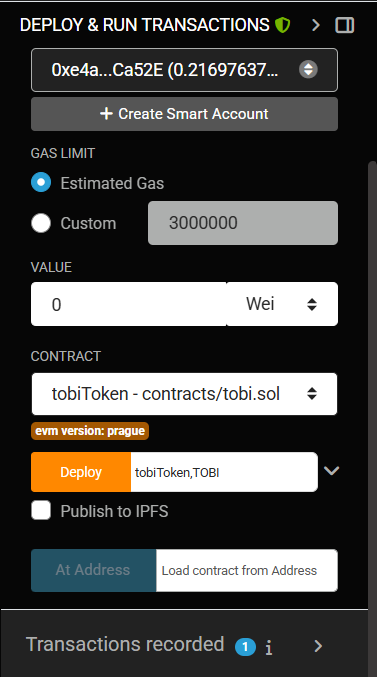
1. Go to Solidity Compiler tab then Select version 0.8.20 then Click Compile.

**Note:** Ensure **no errors** appear in console.

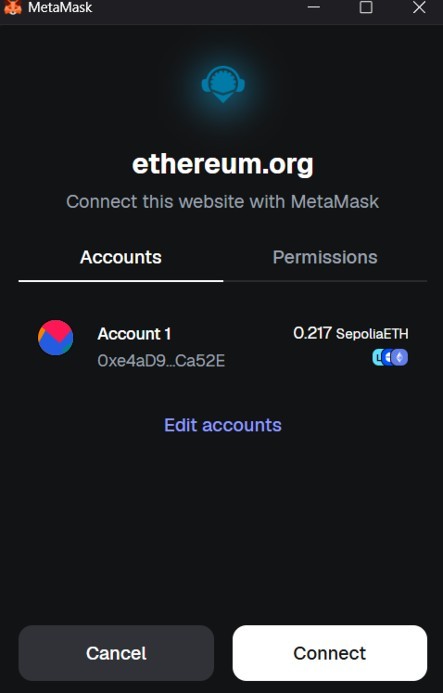


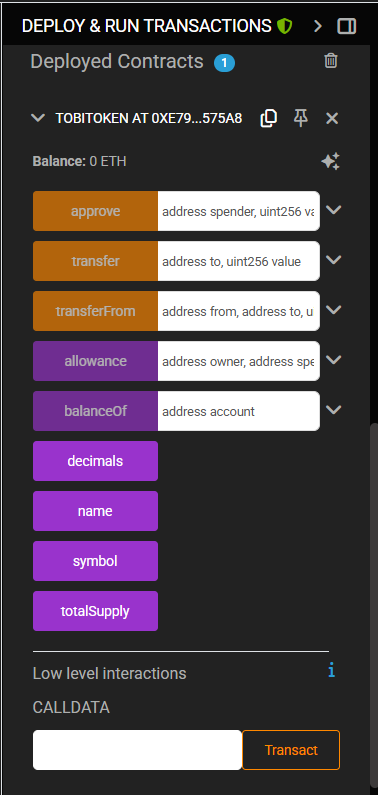
1. Then we go to Deploy & RunTransactions tab.

* Select **Injected Provider – MetaMask** and approve connection.

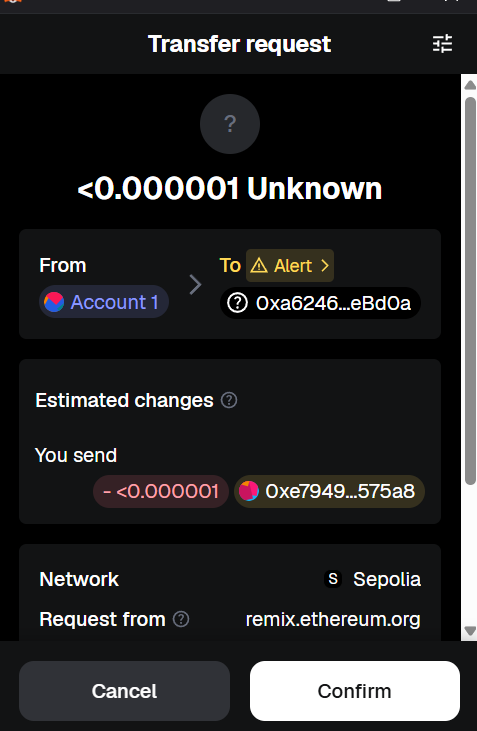
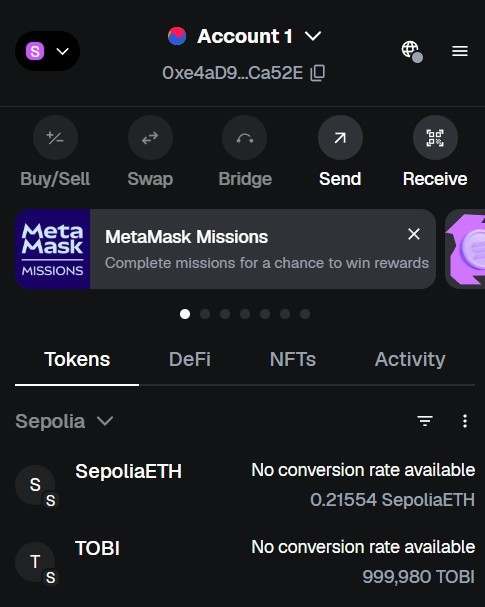


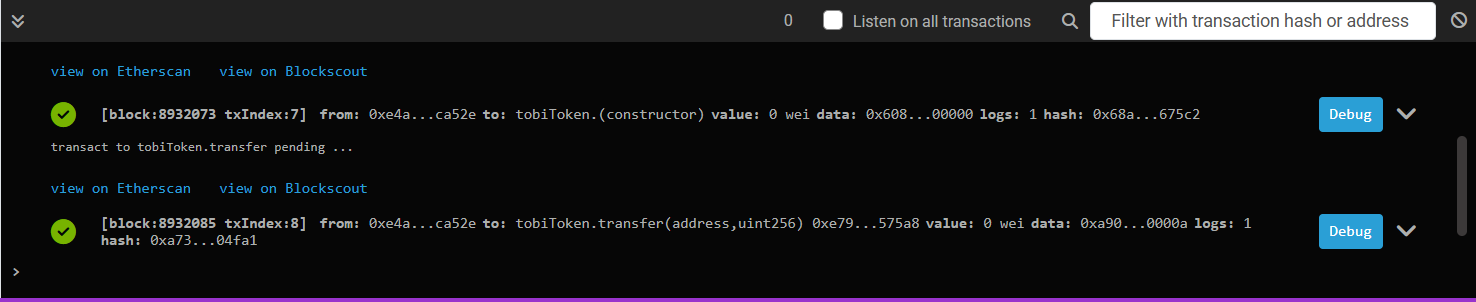
1. Enter **Token Name** and **Symbol** in constructor (e.g., *TobiCoin*, *TOB*).

* Click **Deploy**, approve the MetaMask transaction.
* Copy the **contract address** → Search it on **Etherscan**.

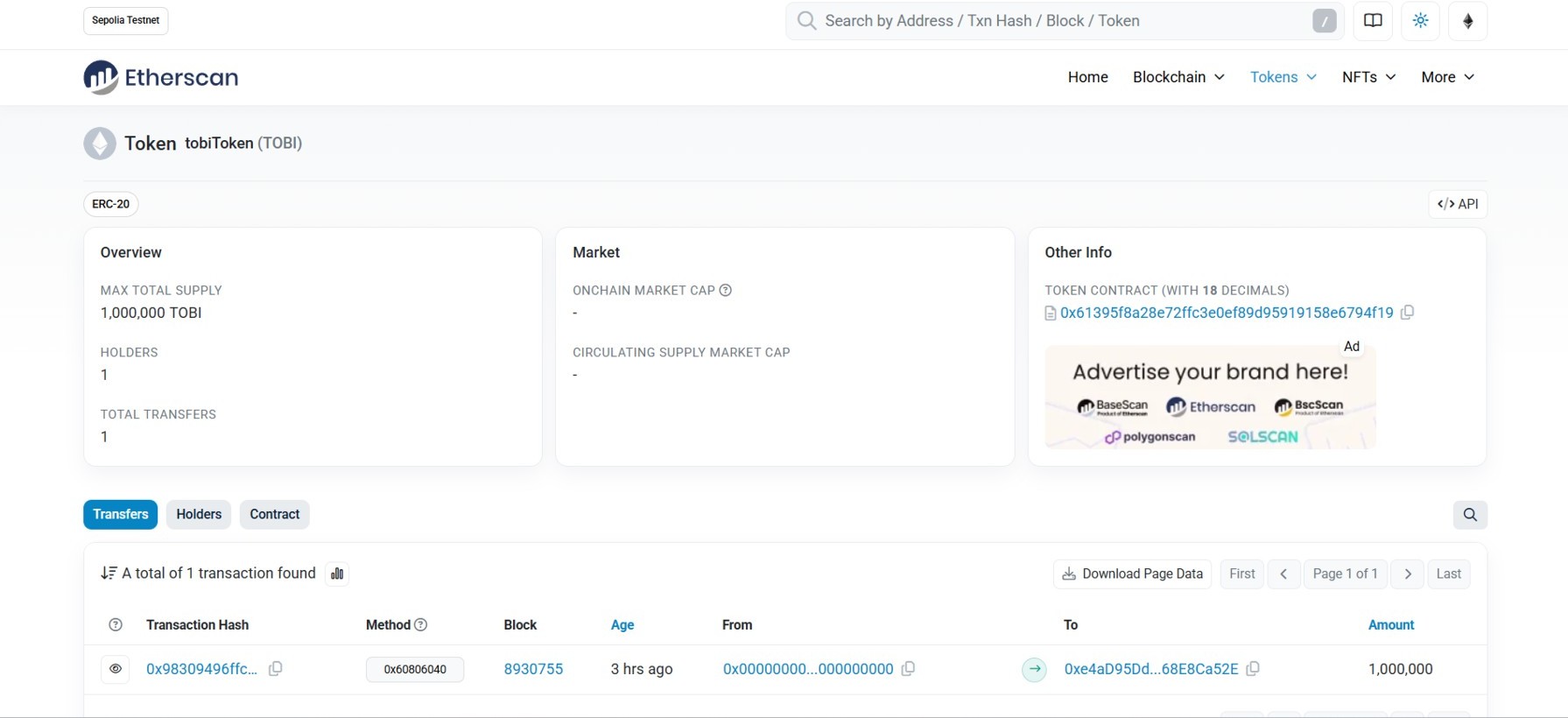


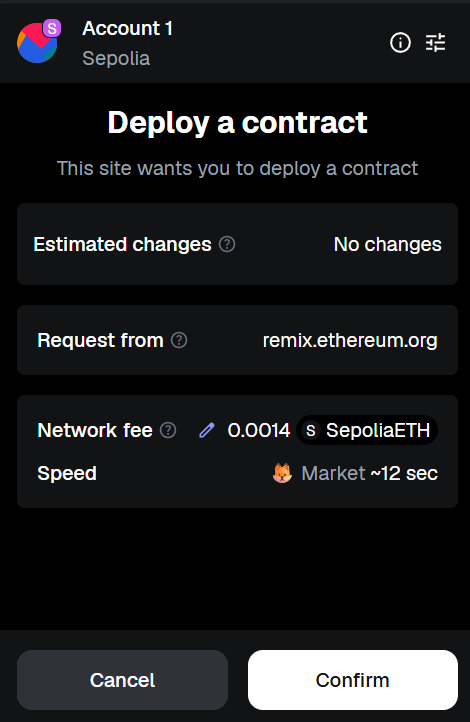
* Open MetaMask → **Import Token** → Paste contract address → Confirm.
* Check your wallet balance (1,000,000 TOB minted).
* In Remix → Expand **Deployed Contract** → Call **transfer(address, amount)** to send tokens.
* Verify the transaction on **Etherscan**.





1. Open [Etherscan](https://etherscan.io) or its respective testnet version (e.g., [sepolia.etherscan.io](https://sepolia.etherscan.io)). Enter the deployed contract address in the search bar. The explorer will display the contract details such as token name, symbol, total supply, transaction history, and contract code for verification.





1. Add Token to MetaMask:

Open MetaMask then Click on "Import Tokens".

Paste the contract address to MetaMask will auto-fill token symbol and decimals.

Click "Add Custom Token" Then "Import Tokens".

You will now see your token balance (1,000,000 TOB) in MetaMask.

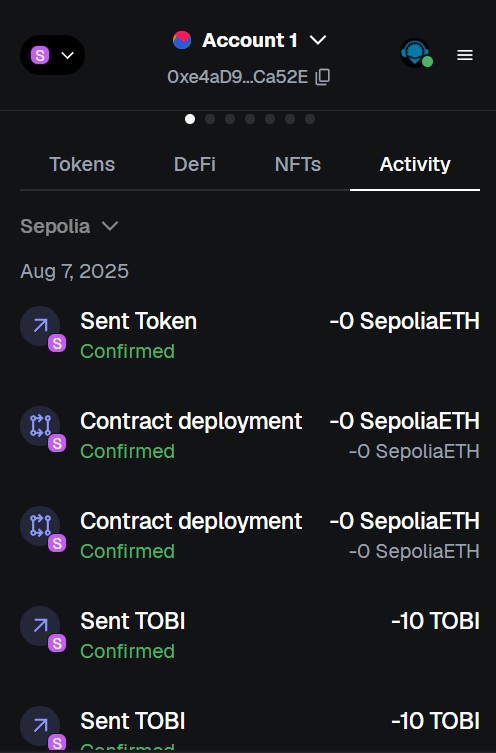
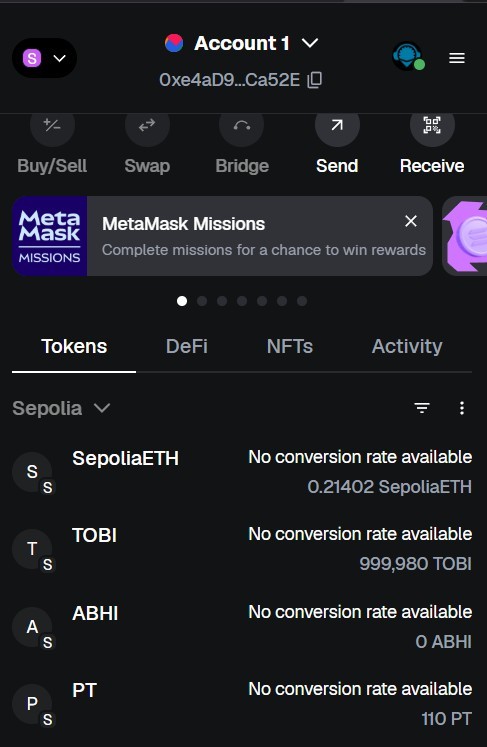
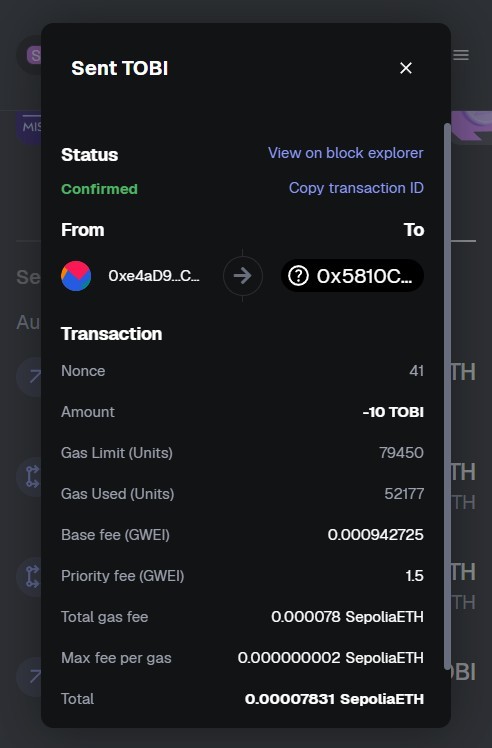
1. ERC-20 token contract compiled successfully without errors.

* Deployment was successful on Ethereum (testnet).
* Token appeared in MetaMask wallet after importing.

Token transfer between two wallets was executed successfully and verified on Etherscan

## \* Implementation Phase: Final Output (no error)

Applied and Action Learning



# Observation:

1. The smart contract was deployed successfully using **Remix + MetaMask.**
2. Tokens were minted automatically to the deployer’s wallet.
3. Imported token appeared correctly in **MetaMask balance.**
4. Transactions (minting & transfer) were visible on **Etherscan,** proving functionality.



|  |  |  |  |
| --- | --- | --- | --- |
| **Rubrics** |  |  |  |
| 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:***



***Signature of the Faculty:***