



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 Experiment :

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

Algorithm:

1. Open Remix IDE.
2. Create a new Solidity file.
3. Write the ERC-20 token contract code.
4. Compile the contract.
5. Select "Injected Provider - MetaMask" as environment.
6. Deploy the contract using MetaMask.
7. Confirm the transaction in MetaMask.
8. Copy the deployed contract address.
9. Explore the token on Etherscan.
10. Add the token to MetaMask using the contract address.
11. Go to Remix, open the deployed contract in the "Deployed Contracts" section.
12. Call the transfer function to send tokens to another wallet.

### \* Softwares used

1. Remix IDE
2. MetaMask
3. Etherscan
4. OpenZeppelin Contracts
5. Brave Web Browser

Page No.....

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

## \* Testing Phase: Compilation of Code (error detection)

Open your browser and go to <https://remix.ethereum.org>. This is where you'll write, compile, and deploy your smart contract. In the left panel, click on the "contracts" folder.

Click the "+" icon to create a new file (e.g., tobi.sol) Write the ERC-20 Token Code

```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract amittoken is ERC20, Ownable {
    constructor(string memory name, string memory symbol) ERC20(name, symbol) {
        _mint(msg.sender, 1000000);
    }
}
  
```

This code creates an ERC-20 token with customizable name and symbol. It also mints 1 million tokens to the deployer's wallet.

Compile the Contract Go to the "Solidity Compiler" tab on the left. Select compiler version 0.8.20. Click the "Compile amittoken.sol.sol" button. Ensure no errors are shown in the console.

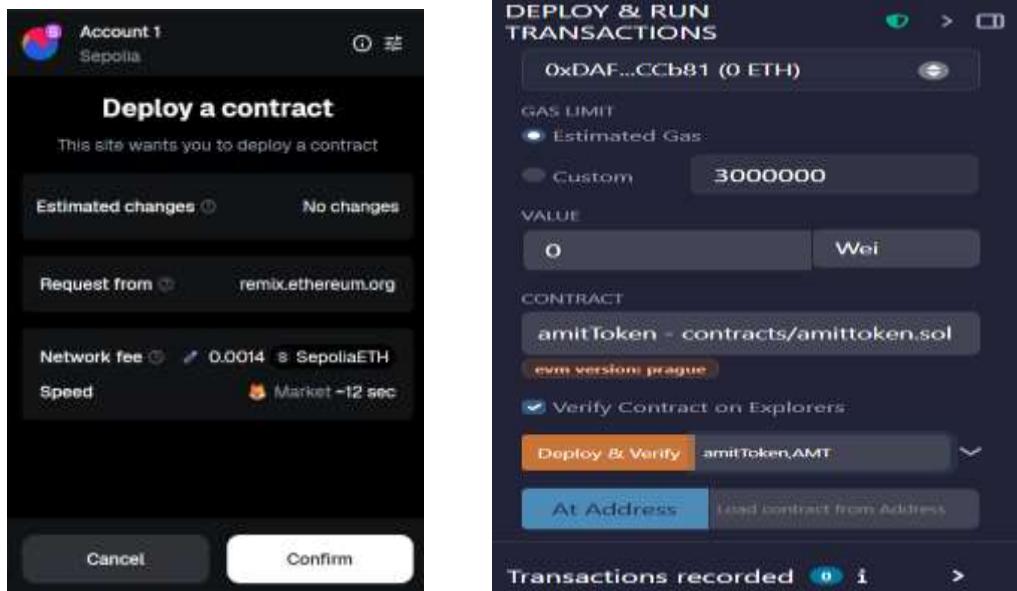


Go to "Deploy & Run Transactions" in Remix. Select "Injected Provider - MetaMask" as the environment. Approve the connection to Remix from MetaMask.



## \* Testing Phase: Compilation of Code (error detection)

Deploy the Contract .Enter the name and symbol for your token in the constructor parameters (e.g., "TobiCoin", "TOB").Click "Deploy".MetaMask will pop up – approve the transaction.



Explore Contract on Etherscan ,Go to <https://etherscan.io> or the testnet version (e.g., [sepolia.etherscan.io](https://sepolia.etherscan.io)).Paste your contract address into the search bar.View details like token name, total supply, transactions, and contract code.

Overview		More Info		Multichain Info	
ETH BALANCE 0.0005145074886588687 ETH		TRANSACTIONS SENT Latest: 2 days ago ↗ First: 98 days ago ↗		N/A	
TOKEN HOLDINGS \$0.00 (0 Tokens)		PAID OUT BY 0x7E5dA5fA...032592ed5f (100 days ago)			

Add Token to MetaMask. Open MetaMask → Click on "Import Tokens". Paste the contract address. MetaMask will auto-fill token symbol and decimals. Click "Add Custom Token" → "Import Tokens". You will now see your token balance (1,000,000 TOB) in MetaMask.

## \* Testing Phase: Compilation of Code (error detection)

The screenshot shows the MetaMask wallet interface connected to Account 1 (0xDAFe4...CCbF). The balance is 0.095 SepoliaETH. There are buttons for Buy/Sell, Swap, Bridge, Send, and Receive. A 'MetaMask Missions' notification is visible, encouraging users to complete missions for rewards.

Tokens	DeFi	NFTs	Activity
SepoliaETH SepoliaETH No conversion rate available 0.09503 SepoliaETH			
AMT AMT No conversion rate available 1,000,000 AMT			

Transfer Tokens to Another Wallet ,In Remix, expand your deployed contract in "Deployed Contracts".Find the transfer function.

The screenshot shows a transaction review screen in the Remix IDE. The transfer amount is 0.01 SepoliaETH from Account 1 to Account 2. The network selected is Sepolia. The network fee is 0 SepoliaETH, and the speed is Market (~12 sec). The 'Confirm' button is highlighted.

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

Applied and Action Learning

The left screenshot shows a transaction history from Etherscan. It details a transfer of -100 AMT SepoliaETH from account OxDADe4...C... to Account 1. The transaction includes gas fees and a priority fee. The right screenshot shows the MetaMask wallet interface for Account 1, which now holds 0.0951 SepoliaETH.

Sent AMT	x
Status	Confirmed
From	OxDADe4...C...
To	Account 1
Transaction	
Nonce	7
Amount	-100 AMT
Gas Limit (Units)	45427
Gas Used (Units)	27489
Base fee (GWEI)	0.000600251
Priority fee (GWEI)	1.5
Total gas fee	0.000041 SepoliaETH
Max fee per gas	0.000000002 SepoliaETH
Total	0.00004125 SepoliaETH

0.0951 SepoliaETH		
+ \$0 (+0.00%)	Portfolio	
<a href="#">Buy/Sell</a>	<a href="#">Swap</a>	
<a href="#">Bridge</a>	<a href="#">Send</a>	
<a href="#">Receive</a>		
<b>MetaMask MISSIONS</b>		
Complete missions for a chance to win rewards		
<a href="#">Tokens</a> <a href="#">DeFi</a> <a href="#">NFTs</a> <a href="#">Activity</a>		
Sepolia		
<a href="#">S</a>	<a href="#">SepoliaETH</a>	No conversion rate available
<a href="#">S</a>		0.09507 SepoliaETH
<a href="#">A</a>	<a href="#">AMT</a>	No conversion rate available
<a href="#">A</a>		1,000,000 AMT

## \* Observations

1. The ERC-20 token contract was successfully compiled and deployed using Remix and MetaMask.
2. The token appeared in MetaMask after importing the contract address, confirming successful minting.
3. Token transfer to another wallet was executed and verified on Etherscan, confirming proper contract functionality.

## ASSESSMENT

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		
<b>Total</b>	<b>50</b>		

**Signature of the Student:**

Name :

Regn. No. :

Page No.....

**Signature of the Faculty:**

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