

  **ERC-20 Basics – Tokenization Concepts**

**Objective/Aim:**

To understand the ERC-20 token standard, its role in enabling standardized token creation, and how it facilitates tokenization, interoperability, and DeFi applications on Ethereum

**Apparatus/Software Used:**

* Smart contract development tools (Remix IDE, Truffle, Hardhat)
* MetaMask
* Blockchain explorers (Etherscan)

**Theory/Concept:**

ERC-20 is a technical standard on Ethereum for creating fungible tokens—meaning each unit is identical and interchangeable. It specifies a set of required functions (e.g., totalSupply, balanceOf, transfer, transferFrom, approve, allowance) and optional metadata (name, symbol, decimals), ensuring compatibility among tokens and Ethereum applications. This standard made large-scale tokenization, DeFi protocols, and token sales possible by guaranteeing that tokens work across wallets, exchanges, and dApps.



**Procedure:**

* Write a smart contract in Solidity that implements the ERC-20 interface, including all required functions.
* Deploy the contract to an Ethereum blockchain network.
* Use wallets and dApps to mint, transfer, or approve spending of the ERC-20 token.
* Leverage blockchain explorers to verify token creation, balances, and transactions.

**Observation:**

* ERC-20 tokens have become the foundation for thousands of projects—ranging from cryptocurrencies, stablecoins, and governance tokens to tokenized assets. Their standardization ensures rapid development, low friction for integration, and high liquidity, even as token functionality can be extended via smart contracts and composable DeFi protocols.