1. Please describe (step by step) how to code a smart contract in Ethereum Blockchain.

I follow the ERC-20 contract interface whereby I implement the needed functions and events. ERC-20 is a proposal of standards to create and exchange tokens on Ethereum network. A smart contract is only ERC-20 compliant when they implement ERC-20 standards.

ERC-20 is not the only standard that exists but it is widely adopted and allows cryptocurrencies or digital assets to be traded more easily on Ethereum. If there aren’t any common standards, the digital wallet must implement different function calls to talk to each of underlying smart contracts which operate trades upon corresponding tokens/assets. A common standard such as ERC-20 ensures that Ethereum wallets or DApps only need to implement once a set of function calls in order to interact with smart contracts of similar nature.

An ERC20 compliant smart contract must contain the functions stated below:

* totalSupply() function returns the total amount of tokens exist on the contract at the moment of query.
* balanceOf() function returns amount of tokens owned by an account at the moment of query.
* allowance() function returns amount of tokens authorized to be withdrawn from a holder’s account by somebody acting on his behalf.
* approve() function allows a holder of coins to grant another account the right to withdrawal coins from his pocket within a specific amount. This function emits Approval event when execution succeeds.
* transfer() function allows to transfer tokens from one’s pocket to another’s. This function emits Transfer event when execution succeeds.
* transferFrom() function allows a representative to transfer coins (which he does not own but has been granted the right to spend), to another account.

3) 

4)

