Blockchain:- A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralized database managed by multiple participants is known as Distributed Ledger Technology (DLT).

Smart Contract :-A "smart contract" is simply a program that runs on the Ethereum blockchain. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.

Smart contracts are a type of Ethereum account. This means they have a balance and can be the target of transactions. However they're not controlled by a user, instead they are deployed to the network and run as programmed. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts can define rules, like a regular contract, and automatically enforce them via the code. Smart contracts cannot be deleted by default, and interactions with them are irreversible.

ERC20:-ERC-20 is the technical standard for fungible tokens created using the Ethereum blockchain. A fungible token is one that is interchangeable with another token—where the well-known non-fungible tokens (NFTs) are not interchangeable.
ERC-20 allows different smart-contract enabled tokens a way to be exchanged. Tokens, in this regard, are a representation of an asset, right, ownership, access, cryptocurrency, or anything else that is not unique in and of itself but can be transferred.

Event:-Event is an inheritable member of a contract. An event is emitted, it stores the arguments passed in transaction logs. These logs are stored on blockchain and are accessible using the address of the contract till the contract is present on the blockchain. An event generated is not accessible from within contracts, not even the one which has created and emitted them.
Mapping:-Mapping in Solidity acts like a hashtable or dictionary in any other language. These are used to store the data in the form of key-value pairs, a key can be any of the built-in data types but reference types are not allowed while the value can be of any type. Mappings are mostly used to associate the unique Ethereum address with the associated value type.

Constructor:- Constructor is a special function declared using the constructor keyword. It is an optional function and is used to initialize state variables of a contract. Following are the key characteristics of a constructor.

- A contract can have only one constructor.
- A constructor code is executed once when a contract is created and it is used to initialize contract state.
- After a constructor code is executed, the final code is deployed to the blockchain. This code includes public functions and code reachable through public functions. Constructor code or any internal method used only by constructor are not included in final code.
- A constructor can be either public or internal.
- An internal constructor marks the contract as abstract.
- In case, no constructor is defined, a default constructor is present in the contract.

Modifier:- Function Modifiers are used to modify the behavior of a function.
The function body is inserted where the special symbol "_;" appears in the definition of a modifier. So if the condition of the modifier is satisfied while calling this function, the function is executed and otherwise, an exception is thrown.

View:-In Solidity, a function that only reads but doesn't alter the state variables defined in the contract is called a View Function.

Pure:- Pure functions ensure that they do not read or modify the state. A function can be declared as pure
External:- External functions are meant to be called by other contracts. They cannot be used for internal calls. To call an external function within the contract this. function_name() call is required.

private:-Private functions/ Variables can only be used internally and not even by derived contracts.

Memory:- The Solidity Smart Contract can use any amount of memory during the execution but once the execution stops, the Memory is completely wiped off for the next execution.

Storage:- Storage is like database data stored in a blockchain node file system. It is persistent and has access to multiple executions of the same contract.

The storage contains key and value pairs, each key and value pair stores 32 bytes of data.

Global variables declared in the global scope of the contract are stored here And Also, Reference type local variables such as Struct, Array, and mapping type are stored in memory by default.

Virtual Override:-
Solidity lets developers change how a function in the parent contract is implemented in the derived class. This is known as function overriding. **The function in the parent contract needs to be declared with the keyword virtual to indicate that it can be overridden in the deriving contract**

Internal:- Those functions and state variables can only be accessed internally (i.e. from within the current contract or contracts deriving from it), without using this

Msg.sender:- the address that has called or initiated a function or created a transaction.

abstract contract:- Abstract contracts are contracts that have at least one function without its implementation or in the case when you don't provide arguments for all of the base contract constructors. Also in the case when we don't intend to create a contract directly we can consider the contract to be abstract. An instance of an abstract cannot be created. Abstract contracts are used as base contracts so that the child contract can inherit and utilize its functions. The abstract contract defines the structure of the contract and any derived contract inherited from it should provide an implementation for the incomplete functions, and if the derived contract is also not implementing the incomplete functions then that derived contract will also be marked as abstract. An abstract contract is declared using the abstract keyword.

Interface:- Interfaces are similar to abstract contracts and are created using interface keywords. Following are the key characteristics of an interface.

- Interfaces can not have any function with implementation.
  - Functions of an interface can be only of type external.
  - Interfaces can not have constructor.

- Interfaces can not have state variables.

Address:- An address is a 20 bytes data type. It is specifically designed to hold account addresses in Ethereum, which are 160 bits or 20 bytes in size. It can hold contract account addresses as well as externally owned account addresses. Address is a value type and it creates a new copy while being assigned to another variable.

Bytes:- In Solidity, byte refers to 8-bit signed integers. Everything in memory is stored in bits with binary values 0 and 1. The bytes value type in Solidity is a dynamically sized byte array. It is provided for storing information in binary format.