VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Internal Assessment Test

Branch	Test Date	Semester	Div.	Roll No.	Student's Signature
CMPN.		IV			

~~IA Test No.~~	Subject
MSB-1 solution	BLOCKCHAIN. (Faculty Solution).

Junior Supervisor's full signature with date :

Question No.	1	2	3	Total 20	Examiners Signature	Student's Sign After receiving the assessed answer sheet
Marks obtained						

Q. 1.

Ch 2

Pros and cons of blockchain.

1. Decentralized & Distributed.

Pros:-

1. Removes single point of failure
2. Fosters transparency, faster consensus.
3. More engaging as everyone is involved in decision.

Cons:-

1. Sometimes traditional db is more suitable to do work faster & cheaper.
2. Stronger players can take control of the n/w.

2. Trustlessness:-

Pros:-

1. Allows multiple entities who do not trust each other to transact with one another.
2. Ensures valid and accurate data.
3. Disintermediation.

Cons:-

1. Every node needs to run the blockchain to verify transactions & maintain consensus.
2. Nodes may prioritize transactions.

3. Immutability.

Pros:-

1. Contains verifiable records of all Tx.
2. No double spending.
3. There is provenance.

Cons:-

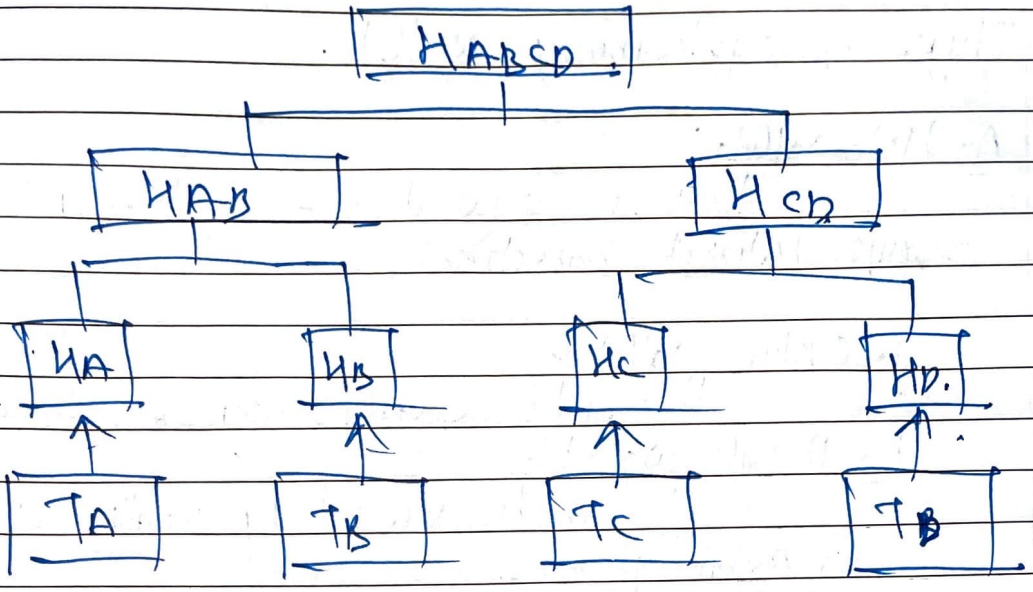
1. Not every node has the capacity to maintain & run a full node copy of BC.
2. 51% attack.

B. Inheritance:-

- Inheritance is a way to extend functionality of contract. Solidity supports both single as well as multiple inheritance.
- A derived contract can access all non-private members including internal methods & state variables.
- Function overriding is allowed provided function signature remains same.
- We can call a super contract's function using super keyword or using super contract name.
- In case of multiple inheritance, function call using super gives preference to most derived contract.

C. Merkle Tree:-

- Merkle tree enhance the efficiency and security of blockchain networks by organizing transaction data in a hierarchical structure that allows for quick and efficient verification.
- Each transaction is hashed and these hashes are paired & hashed together, forming a tree where the root hash, known as Merkle root, represents all the transactions.



- This structure ensures that any change in the transaction data will alter the merkle root, making it easy to detect tampering.
- Additionally, merkle trees allow blockchain to efficiently verify the integrity of large no of transactions without storing all the data, saving space and computational resources.
- This makes them a key component in maintaining the security & scalability of LC n/w.

Q2.

a. UTXO.

- An unspent transaction output (UTXO) is the technical term for the amount of digital currency that remains after cryptocurrency transaction.
- It is a amount of digital currency someone has left remaining after executing a Tx.
- When a Tx is completed, the unspent output is deposited back into database as input which can be used later for another Tx.
- UTXOs are created through a consumption of existing UTXOs. Every BTC transaction is composed of inputs & outputs. Inputs consume an existing UTXO while o/p create new UTXO.

b. Types of cryptocurrency wallets.

A. Hot wallets:
They are accessed for day-to-day Tx and require internet connection.

a. Online wallet:
It can be accessed using web browser

b. Mobile wallet:
It can be accessed by installing applications in mobile phone.

c. Desktop wallet:
It can be accessed by installing desktop applications

B. Cold wallet.

a. Hardware wallet:
In this private key can be set into hardware & it is used to be plug while accessing wallet.

b. Paper wallet:
In this, paper QR code will be generated with keys and amount.

c. Error handling functions
Following are some of the important methods used in error handling

1. assert(bool condition)
In case condition is not met, this method call causes an invalid opcode and any changes done to state get reverted. This method is to be used for internal errors.

2. require(bool condition)
In case of condition not met, this method call reverts to original state. This method is to be used for errors in inputs or external components.

3. require (bool condition, string message)
In case condition is not met, this method call reverts to original state. This method is to be used for errors in inputs or external components. It provides an option to provide a custom message.

4. revert ()
This method aborts the execution & revert any changes done to the state.

5. revert (String memory reason)
This method aborts the execution and revert any changes done to the state. It provides an option to provide a custom message.

Q. 5.
a.
Different Types of BC:-

1. Public BC:-
- It is public permissionless blockchain.
- In public BC, anyone in the world can access the BC, download a copy and run the code.
- One does not need any permission to read/access a Tx, initiate Tx or participate in the consensus process to create a block.

2. Private BC:-
- It is private permissioned BC.
- The info is not open for everyone.
- Features of decentralization and openness is lost as all the permissions are controlled by few nodes in organization.
- Here owner has sole control over who can read, write or validate the data, it stands to reason why many may not consider it to be real BC.

3. Consortium BC:
- It is also known as Federated BC.
- It is combined BC and considered to be hybrid between public and private BC.
- It is a distributed ledger that anyone can download or access and the process is NOT controlled by one company but by the predetermined consortium of companies or representative individuals.

	Public BC.	Private BC.	Consortium BC.
- Accessibility:	Open to all	Restricted	Restricted.
- Users:	Anyone can participate.	Controlled by single organization.	Managed by a group of organizations.
Consensus Algo:	PoW or PoS.	PBFT or Raft	Voting based.
Operations:	Fully transparent and decentralized.	Private data and controlled access.	Partially decentralized with shared control.

6. Limitations and challenges.

1. Coding errors:
   Vulnerabilities in smart contract code can lead to financial losses.

2. Scalability:
   Slow tx times and high fees in high-demand situations.

3. Legal uncertainty:
   Varied legal status and compliance risks across jurisdictions.

4. Immutability:
   Difficulty in updating contracts once deployed.

5. Security:
   Risks of hacks if contracts are not properly secured.

6. Interoperability:
   Challenges in interacting across different BC nlw.

7. Resource Intensity:
   High cost and energy consumption for execution

Impact on adoption:-
These challenges lead to higher costs, security risks, legal uncertainties and integration difficulties which can slow the adoption of BC technology across industries.