Heaven Heritage

Eth (Ether)

apsara

Date: __/__/____

GAS (Gwei): Unit to measure the gas

1 Gwei = 0.000000001 ETH

Gas is the unit used to measure computational worth required for transactions on Smart Contracts and execution on Ethereum block.

* Gas limit: It refers to the maximum amount of Gas that a user is willing to spend on a transaction. Higher complexity operation req. more gas.

* Gas price: It determines how much a user pay's per unit of gas, typically representing in Gwei = 0.000000001 ETH.

Formula of Gas cost calculation:

The total cost of executing a txn is executed as:

Cost = Gas used x Gas price

5 Sheet 20 only BCT

10 marks single sheet

7 sheet 10 2 sides

50/24

apsara

Date:

⇒ Use in ethereum : Gas ensures that resources are efficiently allocated and prevents spam or malicious behaviour by user to pay for each computation.

⇒ Transaction fees : A txn fees is a small amount of cryptocurrency paid by users to process txn on a blockchain network. It is paid to miners or validators as a reward for confirming and securing txn.

In most BC systems like bitcoin and ethereum txn fee depends on network conjetion and data size.

During high demand, fee increase as users complete for the next block.

Data size : Large txn may needed higher fees.

⇒

apsara

Date:

Fees mode in Bitcoin : Fees are based on size of txn in bytes and network demand.

Gas fees are used to calculate txn fees in ethereum blockchain

Solidity : It is high level, object Oriented language designed for writing Smart contracts on the EVM.
It is similar to javascript and python, It provides features to define contracts, variables, functions and data structures.

* How to execute the code in Solidity program.

1) Offline : To operate a Solidity Smart contract in offline mode, it must ~~not~~ following cond.

1)- Download and install node.js
2). Install truffle globally.
3). Install GANACHE-CLI

⇒

Actions :

1). Create a truffle project and set up a development network.
2). Develop and deploy smart contract.
3). From the truffle console interact with smart contract.
4). Create test to evaluate solidity primary features.

2). Online : In online mode, the remix IDE is typically used to compile and run solidity smart contracts.

Datatypes supported by solidity programming :

1). Boolean, Integer, String.

Truffle : It is a popular development framework for ethereum. Used for developing the smart contract, testing and deployment.

⇒

* Smart Contract :

Key features :

1). Development environment
2). Testing framework (provide automated testing)

Manage the deployment of contracts to various Ethereum networks.

Components :

1). Truffle CLI : Command line tool for running task such as compiling contracts.

2). GNACLI + CLL

→ DRIZZLE : Front end library that connects Smart Contracts to web applications.

1). TESTNETS
2). MAINNET → real cryptocurrency used

apsara

Date: / /

Design and issue cryptocurrency:

- Steps to create a cryptocurrency -

Step: 1 Define the purpose and features

1). Utility token (To provide access to the services).

2). Security token (Present ownership)

3). Governance token (allow voters to vote on protocol change).

Step: 2 Choose a blockchain platform:

1) Ethereum
2). Binance Smart chain (offer lower fees)
3). Solana (produce high throughput)

Step: 3 Select a token standard.

1). ERC - 20
2). ERC - 721
3). ERC - 1155

} Diff., Prop., adv., disadv.

⇒

Page No:

Step: 4 Develop the smart contract :

→

Step: 5 Security audit :

1). Code review
2). Automated tools. → Myth X

Step: 6 Deploy the token

1). TESTNET development (tool - Ropsten)
2). MAINNET development {

Mining :

Def.
Process.

⇒ Mining algorithms :

1) SHA-256
2) ETH - Hash
3) QUARK
4) Present key algo. } less computational time as
5) Twin algo. } compared to ~~256~~ SHA-256.

⇒

apsara

Date:

Mining hardware :

1). CPU mining
2). GPU mining
3). ASIC

Environment impact :

Bitcoin Ethereum

DAPPS (Decentralised APPS) : These are applications that run on decentralised N/W using smart contract for their backend logic blockchain for data storage.

Characteristics :

1). Open Source
2). Decentralised backend (operates on P2P N/W)
3). Token
4). Architecture
   - 1. Frontend
   - 2. Smart Contract (Backend logic)
   - 3. Blockchain network.

apsara

Date: __ - __ - ____

categories of DAPPS :

1. Defi (Decentralised finance) Used by U.S bank.
2. For gaming pupose
3. Social media.
4. Supply chain (Amazon, Walmut).

Development tools :

1). Meta Mask
2). Web. 3. JS
3). IPFS (Used in big data).

Challanges :

1). Security
2). Scalability

apsara

Date: / /

Unit-3

Bitcoin and consses algorithm :

Bitcoin Consses : mechanism is a System that cryptocurrency like Bitcoin and Ethereum used to validate the authenticity of transaction and maintain the security of the underlying blockchain.

A consensus mechanism is a self regulatory stack of protocal written in a blockchain code synchronise a network about the state of a digital ledger.

Blockchain Consensus algo. : It ensure each new block added to the network is the only version of truth, which is agree by all the nodes in a decentralised computing network.

Types of consensus mechanism :

1). POW (Work)
2). POS (Stack)
3). POA (Authority)
4). Dedicated POS
5). Proof of Capacity
6). Proof of Activity
7). Proof of Elapse type

apsara

Date: / /

8). Proof of Burn

Hash & cash PoW : (used for e-mail).
Hash cash wash a sol. designed to combat spam by generating a pow that allow verify that a sudden e-mail was not spam. It is used as a denial of service counter measure technique in a no. of systems.

2 Marks

Bitcoin Proof of work :

Attacks on PoW :
1). Distributed Denial of Service (DDOS).
2). Sybil attack (1 user create multiple ID and retrieve inf)

Blockchain Technology

Q1 # Permissionlised model and design use case.

Q2 # Design issues for Permissionised Blockchain

Answer

Ans-2 Permission Blockchain :

It is a part of blockchain technology which is a part of private blockchain.

In this blockchain there is high security because there are limited amount of users and designed for a specific purpose and only permissioned users has the right to access the blockchain.

Issues in designing :

1). Restricted users or limited access :

In this it has a limited access for the users it is good in terms of security.

⇒

Scalability is adv. or dis. — (2) M

apsara

Date: ___/___/___

Wells Fargo → U.S Bank

2) Centralised : It is controlled and managed by a central authority and can monitor request of all users or client.

3) Designed for specified use of Private Organizations:

---

2) Permissioned Blockchain design issues :

1. Access Control
→ Consensus mechanism
→ Scalability
→ Privacy
→ Regulatory Compliance
→ Integration

apsara

Date: __/__/____

Smart Contracts : It is an agreement between two people and activities in a form of computer code. program to execute automatically.

These are executed of Blockchain which means that the terms are stored in a distributed database and cannot be changed.

How does a smart contract work?

The smart contract is similar to other blockchain transfers.

Initial steps :

→ A User initiates transaction.

→ The Txn arrives at distributed database where the identity is confirmed.

→ The Txn which may be transfer of funds approved.

→ The Txn includes the code that defines what type of Txn is to be executed.

→ The Txn are added as block & within blockchain.

→ Any change in contract status step to updated.

Page No. _____

Smart Contract platforms :

1). Ethereum - These are written in programming language called : Solidity and executed by EVM.

Hyper ledger an open source develop-ed by Linux foundation that is not a cryptocurrency but a flexible platform from which smart contract can be developed.

2). Counter party - It use the cryptocurrency BC and allow smart contracts to be developed on it. It is Basically used in bitcoin txn.

→ Polka Dot : It is an alternative to blockchain and is famous for its ability to host para chains (chains within chains), allowing more txn than usual.

⇒ Application :

1). Reward
2). Trade
3). Supply chain
4). Mortgages
5). Property market
6). Health
7). Election
8). Insurance

Proof of Stake vs PoW vs Po Elapswan

Section - 8

apsara DOMS

Date

Consesus models for permission blockchain

Types Consesus protocals for "

1) Pbft (Practical Byztyne fault tolerance).
   ↳ Low latency.
2) - - - - - → Federated Consesus.
3) - - - - - → Round Robin Consesus
4) Proof of Authority (PoAH).

Distributed Consesus in Enclosed Environment:

Elapswan : It is a blockchain-based project that aims to provide a platform for decentralized finance (Defi) and digital asset management. It focuses on enabling users to create, trade, and manage digital assets securely and efficiently.

Key features often associated with such platform include:

1) Smart contracts : Automated contracts that execute transactions based on predefined conditions.

2) Decentralization : Reducing reliance on centralized entities, enhancing security and transparency.

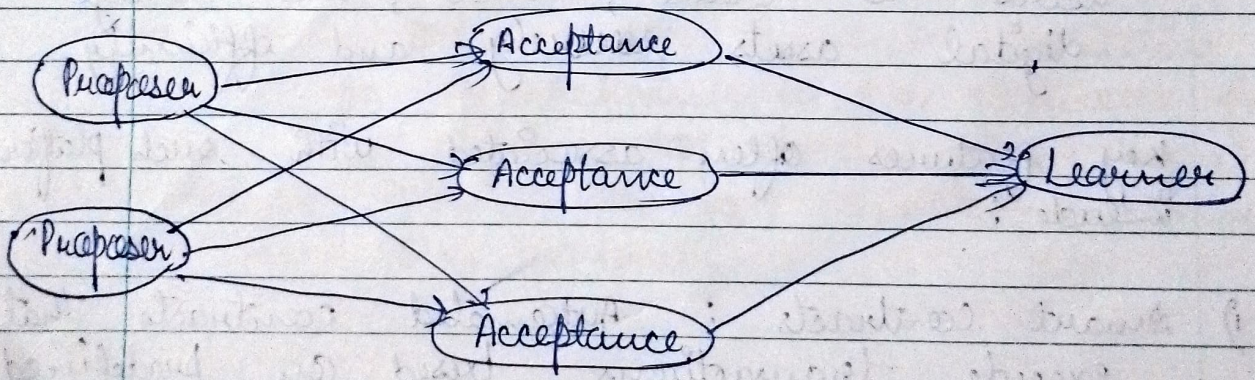
Page No.

3). Interoperability : Enabling different blockchains to communicate and share data.

4). User Empowerment : Giving users control over this assets and financial activities without intermediaries.

Distributed Consensus in enclosed environment :

PaXoS : 3 Concepts :
1). Proposer
2). Acceptance
3). Learner



Phase 1). Prepare
Phase-2). Promise
Phase-3) Accept
Phase-4).

features
1). Safety
2).

apsara

Date: __/__/____

The PaXoS algo. operates in a series of phases ensuring that a group of nodes reach a consensus on a single value. The consensus process is divided into 3 Phases:

1). Prepare
2). Promise
3). Accept

These phases involve message exchange b/w proposer and acceptors to reach an agreement of a value.

1). Prepare Phase: A proposer initiates the consensus process by sending a prepare message with a proposer no.

The proposer no. is a unique identifier for the new proposer and helps prevent conflicts b/w competing proposals.

→ Upon Receiving the prepare messages each acceptor checks if the proposer no. is greater than any prev. proposer no. then acceptor replies with a promise note to accept any proposer

Flowchart

with lower no. and includes info. about the highest no. proposal it has accepted.

2). Promise Phase : After receiving prepare message acceptors response with promise message with proposal.

3). Accept : If a proposal receives promise from a majority of acceptors it includes onto the accept phase.

4). Learn : The learn phase ensure that decision is reliability to communicated to all nodes in a system. learners and proposers. so that the distributed system achieve agreement on the user value.

Property of Paxos :

1) Safety
2) Liveness