

Grp - C

## Experiment - 1

Aim:- Installation of Metamask and study Ether pending per transaction.

Objective:- Understand and explore working of blockchain technology and its application.

Course outcome:- Interpret the basic concepts in blockchain technology and its application.

Description:-

- Metamask is popular cryptocurrency wallet known for its ease of use, availability on both desktops and mobile devices, the ability to buy, send and receive cryptocurrency from within the wallet and collect non-fungible tokens (NFTs) across two blockchain.
- While experienced crypto users will appreciate the simplicity and fast transaction, those new in space are at higher risk of losing their tokens from lost seed phrases, malicious websites and other cryptocurrency system scams.

## Crypt - Experiment - 2.

Aim:- Create your own wallet using Metamask for crypto fan.

Objectives:- Understand and explore the working of BCT and its appl".

Course outcome:- Interpret the basic concepts in BCT and its appl".

Description:-

MetaMask is free crypto wallet software that people can use to interact in the crypto world. It lets you buy, sell, and trade crypto assets for the Ethereum blockchain, much like how real wallet lets you purchase items in real world. It's been around since 2016. ConsenSys, the largest Ethereum development company in world, launched it as a tool to simplify access to decentralized appl" (DApps). It is possible on MetaMask to :-

- Buy, receive, send and swap Ether (ETH), the main token on Ethereum.
- Buy, receive, send, and swap nonfungible tokens (NFTs) in marketplaces
- Connect to Ethereum deaf DApps.

- Connect to other crypto wallets.
- Play blockchain-based games.
- Access different networks such as the BNB smart chain and other testnets.

MetaMask is free to use and can be installed as an extension on internet browsers, Google Chrome, Firefox, Brave, Edge, or download as a smartphone app both on iOS and android. With over 30M users, MetaMask is one of the most popular cryptocurrency wallets today.

## Graph - Experiment - 4.

Aim :- write a program in solidity to create student data.  
Use the following constructs :- structures, Arrays, fallback.  
Deploy this as smart contract on Ethereum and observe  
the tx fee and gas values.

Objectives:- Understand and explore the working of  
blockchain technology and its appln.

Course outcome:- Interpret the basic concepts in BCT and  
its appln.

Theory:-

Struct :- Structs in solidity allows you to create more  
complicated data types that have multiple properties.  
You can define your own type by creating a  
struct. They are useful for grouping together related  
data. Structs can be declared outside of a contract  
and imported in another contract. Generally, it is  
used to represent a record. To define a structure  
Struct keyword is used, which creates a new  
datatype.

Syntax:-

```
Struct <Structure_name> {  
    <data_type> variable_1;  
    <data_type> variable_2;  
}
```

For accessing any element of struct, dot operator  
is used, which separates the struct variable and

the element we wish to access. To define the variable of structure data type structure name is used.

## 2) Arrays:-

Arrays are data structures that store the fixed collection of elements of same data types in which each and every element has specific location, called index. Instead of creating numerous individual variables of same type, we just declare one array of required size and store the elements in the array and can be accessed using the index. In solidity, an array can be of fixed size or dynamic size. Arrays have a continuous memory location, where the lowest index corresponds to the first element while the highest represents the last.

### Creating array:-

Syntax:-

<data type> <array name> [size] = <initialization>.

Fixed size array:- The size of array should be predefined.

Example:- In below example, the contract types are created to demonstrate how to declare and initialize fixed-size arrays.

|| Solidity program to demonstrate.  
|| creating fixed-size array.  
pragma solidity ^0.5.0;

|| Creating a contract.

```
contract Types {  
    || Declaring State variables.  
    || of type array.  
    uint[6] data];
```

|| Defining fun to add values to array.

```
function arrayExample() public returns(  
    int[5] memory, uint[6] memory) {  
    int[5] memory data = [int(50), -63, 77, -28, 90];  
    data1 = [uint(10), 29, 39, 40, 50, 60];  
    return(data, data1);
```

y

y

Dynamic array:-

The size of array is not predefined when it is declared. As the elements are added the size of array changes and at the runtime, the size of array will be determined.

Push:-

Push is used when new element is to be added in dynamic array. The new element is always added at last position of array.

Example:- In the below example, the contract Typos first initializes an array [data], and then more values are pushed into the array.

// Solidity program to demonstrate.

// push operation.

pragma solidity ^0.5.0;

// creating a contract

contract Typos {

// Defining the array

uint data = [10, 20, 30, 40, 50];

// Defining the function to push values to array.

function array - push() public returns (uint[] memory) {

data.push(60);

data.push(70);

data.push(80);

return data;

}

}

Fallback function:-

The Solidity fallback function is executed if none of the functions match the function identifier or no data was provided with the function call. Only one

Unnamed function can be assigned to a contract and it is executed whenever the contract receives plain Ether with no data. To receive Ether and add it to total balance of the contract, the fallback fun<sup>n</sup> must be marked payable. If no such fun<sup>n</sup> exists, the extra contract cannot receive Ether through regular tran and will throw an exception.

Properties of a fall back fun<sup>n</sup>:

1. Has no name or arguments.
2. If it is not marked payable, the contract will throw an exception if it receives plain ether without data.
3. Can not return anything.
4. Can be defined once per contract.
5. It is also executed if the caller meant to call a function that is not available.
6. It is mandatory to mark it external.
7. It is limited to 2300 gas when called by another function. It is so as to make this fun<sup>n</sup> call as cheap as possible.

Example:-

```
pragma solidity ^0.4.0;  
// creating a contract.
```

```
contract gfg
```

```
{
```

```
// Declaring the state variable  
uint i;
```

```
// Mapping of address to their balances.
```

mapping (address  $\Rightarrow$  uint) balance;

// Creating a constructor  
constructor() public

{

// Set x to default value of 10.

x = 10;

}

// Creating a fun

fun setX(uint \_x) public returns(bool)

{

// Set x to the value set

x = \_x;

return true;

}

// This fall back fun will keep all the Ether  
function() public payable.

{

balance[msg.sender] += msg.value;

}

// Creating the sender contract  
contract Sender

{

function transfer() public payable

{

// Address of gfg contract

address receiver = 0xbcc018544106d0F0452D54AE0F6;

11 Transfers 100 Eth to above contract receiver from  
receiveTransfer(100);

y  
3

Implementation:-

Write a program in solidity to create Student data.

Use following constructs:- Structures, Arrays, Built-in functions.  
Deploy this as smart contract on Ethereum and observe  
the txn fee and Gas values.

11 code.

```
pragma solidity ^0.6.0;
contract StudentManagement {
    struct Student {
        int stud_id;
        string name;
        string department;
    }
    Student[] students;
    function addStudent(int stud_id, string memory name,
                        string memory department) public {
        Student memory stud = Student(stud_id, name, department);
        students.push(stud);
    }
    function getStudent(int stud_id) public view returns (string
memory, string memory) {
        for (uint i=0; i < students.length; i++) {
            Student memory stud = students[i];
            if (stud_id == stud.stud_id) {
                return (stud.name, stud.department);
            }
        }
    }
}
```

```
if(stud.stud.id == stud-id) {  
    return(stud.name, stud.department);
```

}

if

```
return("not found", "not found");
```

}

if

Group - C

Experiment - 5

Title :- Survey report on types of Blockchains and its real time use cases.

Aim:- Write a survey report on types of Blockchains and its real time use cases.

Objective:- Students should be able to learn new technology such as metamask, application and implementations.

These are

Theory:-

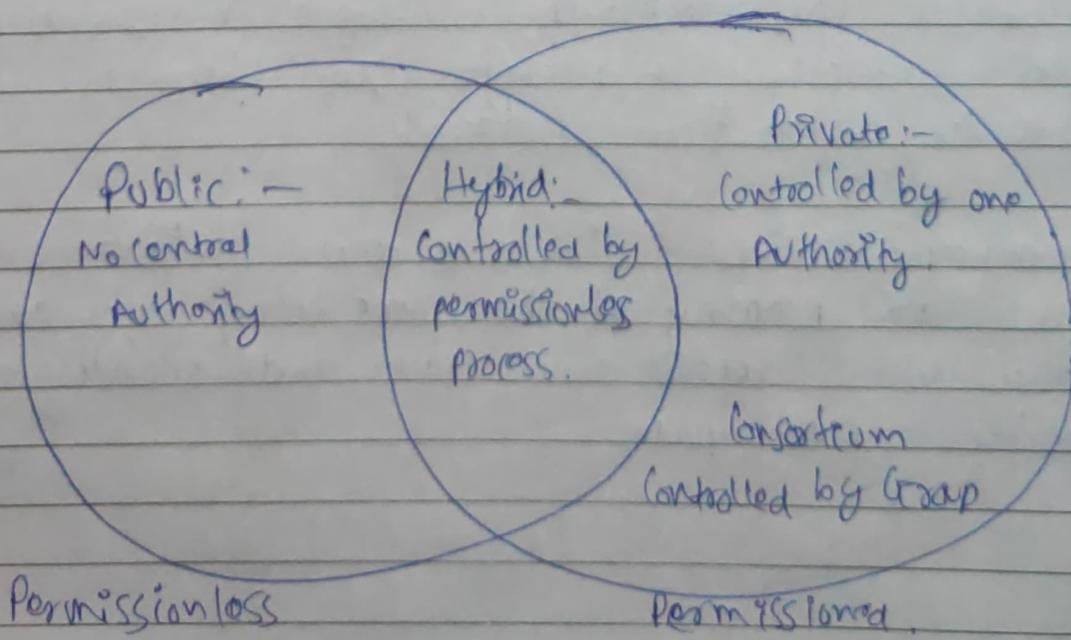
Types of blockchain:-

Public blockchain.

Private blockchain.

Hybrid blockchain.

Consortium blockchain.



## 1. Public blockchain:-

Those are completely open to following the idea of decentralization. They don't have any restrictions, anyone having a computer and internet can participate in network.

As name is public this blockchain is open to public, which means it is not owned by anyone. Anyone having internet with good hardware can participate in this public blockchain.

### Advantages:-

Trustable:- These are algm to detect no fraud. participation need not worry about the other nodes in network.

Secure:- This blockchain is large in size as it is open to public. In a large size, there is greater distribution of records.

Anonymous Nature:- It is a secure platform to make your fin property at the same time, you are not required to reveal your name and identity in order to participate.

## Disadvantages:-

Processing:- The rate of tx process is very slow, due to its large size. Verification of each node is very time consuming process.

• Energy Consumption:- Proof of work is high energy consuming. It requires good computer hardware to participate in the network.

Acceptance:- No central authority is there so govt. are facing the issue to implement the technology faster.

## 2. Private blockchain:-

These are not as decentralized as public blockchain, only selected nodes can participate in process, making it more secure than others.

These are not as open as public blockchain.

They are open to some authorized users only.

These blockchains are operated in closed network.

In this few people are allowed to participate in a network within a company / organization.

## Advantages:-

Speed:- The rate of transaction is high due to small size. Verification of each node is less time-consuming.

Scalability:- We can modify the scalability. The size of network can be decided manually.

Privacy:- It has increased the level of privacy for confidentiality reasons as the businesses required.

Balanced!- It is more balanced as only some users have the access to the ten which improves the performance of network.

## Disadvantages!-

Security:- The no. of nodes in this type is limited so chances of manipulation are there. More Vulnerable.

(centralized):- Trust building is one of the main disadvantages due to its central nature, organization can use this for malpractices.

Count:- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered.

Use cases:- With proper security and maintenance, this blockchain is great asset to secure info without exposing it to public eye. Therefore companies use them for internal auditing, voting and asset management. An example of private blockchains is Hyperledger, Corda.

### 3. Hybrid blockchain:-

Mixed content of private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.

It is combination of both public and private.

Permission-based and permissionless systems are used.

User access info via smart contracts.

Even a primary entity owns a hybrid blockchain it cannot alter the chain.

### Advantages:-

Ecosystem:- Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don't have access to the network.

Cost:- transaction costs are cheap as only a few nodes verify the transaction. All the nodes don't carry the verification hence less computational cost.

Operations:- It can choose the participants in blockchain and decide which transaction can be made public.

### Disadvantages:

Efficiency:- Not everyone is in position to implement hybrid blockchain. The organization also faces some difficulties in terms of efficiency in maintenance.

Transparency:- There is a possibility that someone can hide info from user. If someone wants to get access through a hybrid blockchain it depends on the organization whether they will give or not.

Use case:- It provides a greater solution to health care industry, government, real estate, and financial companies. It provides a remedy where data is to be accessed publicly but needs to be shielded privately.

## 4: Consortium blockchain:-

It is a creative approach that solves the needs of organization. This blockchain validates the tx and also initiates or receives tx.

Also known as Federated Blockchain.

This is an innovative method to solve the organization's needs.

Some part is public and some part is private.

### Advantages:-

Speed :- A limited no. of users make verification fast.  
High speed makes this more usable for organizations.

Authority :- The info of multiple organizations can take part and make it decentralized at entry level. More secure by decentralized authority.

Privacy :- The info is checked blocks is unknown to public view, but any member belonging to blockchain can access it.

## Disadvantages:-

**Approval:** - All members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.

**Transparency:** - It is easier hacked if organization becomes corrupt. Organizations may hide info from uscos.

**Use cases:** - It has high potential in businesses, banks, and other payment processors. Food tracking of the organizations frequently collaborates with their sectors making it federated sol<sup>n</sup> ideal for their use. Examples of consortium blockchain are Tendermint and Multi-chain.

**Conclusion:** - In this way we have explored types of blockchain and its appl<sup>n</sup> in real time.