Peoples Empowerment Group

ISBM COLLEGE OF ENGINEERING, PUNE

Academic Year 2023-24

Department of Computer Engineering

AY 2023-24, Sem-I

Lab Manual (BE Comp)

Name of Subject : **LP-III (BCT)** Class: **BE**

Subject Incharge: **Prof.Shobha Bamane**

Assignment No. 1 : Installation of Metamask and study Spending Ether per Transaction.

Objective : Understand and explore the working of Blockchain Technology and its applications.

Course Outcome : CO6: Interpret the basic concepts in Blockchain technology and its applications.

Theory :

MetaMask:

1. Introduction

MetaMask is a plug-in Ethereum crypto wallet for Chrome onboard users. Available as a browser extension and as a mobile app, MetaMask equips us with a key vault, secure login, and token wallet—everything we need to manage our digital assets. MetaMask provides the simplest yet most secure way to connect to blockchain-based applications.

2. MetaMask Setup

Complete information and study guide about MetaMask can be found at its official website

metamask.io. We need to choose the right browser (Chrome is recommended) and follow its

installation instruction. When we are creating a new MetaMask account, here are some key points we need to pay attention to.

First of all, creating a new strong password is extremely important because it encrypts private key.

Private keys give access to all of our Ether or other tokens. So, it is better to have a strong password here.

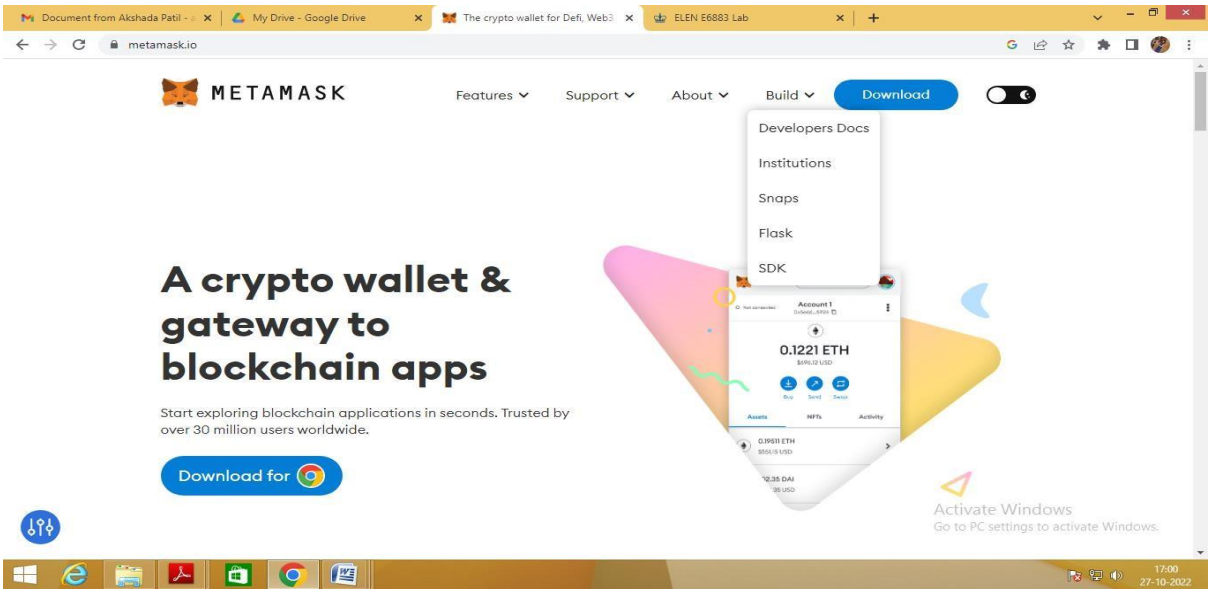
Secret Backup Phrase, which includes 12 mnemonic words, will pop out after setting up the

password. We need to write this phrase on a piece of paper or store it in a secure location because secret backup phrase makes easier to back up and restore our account if we log out our account or accidentally clear browser history.

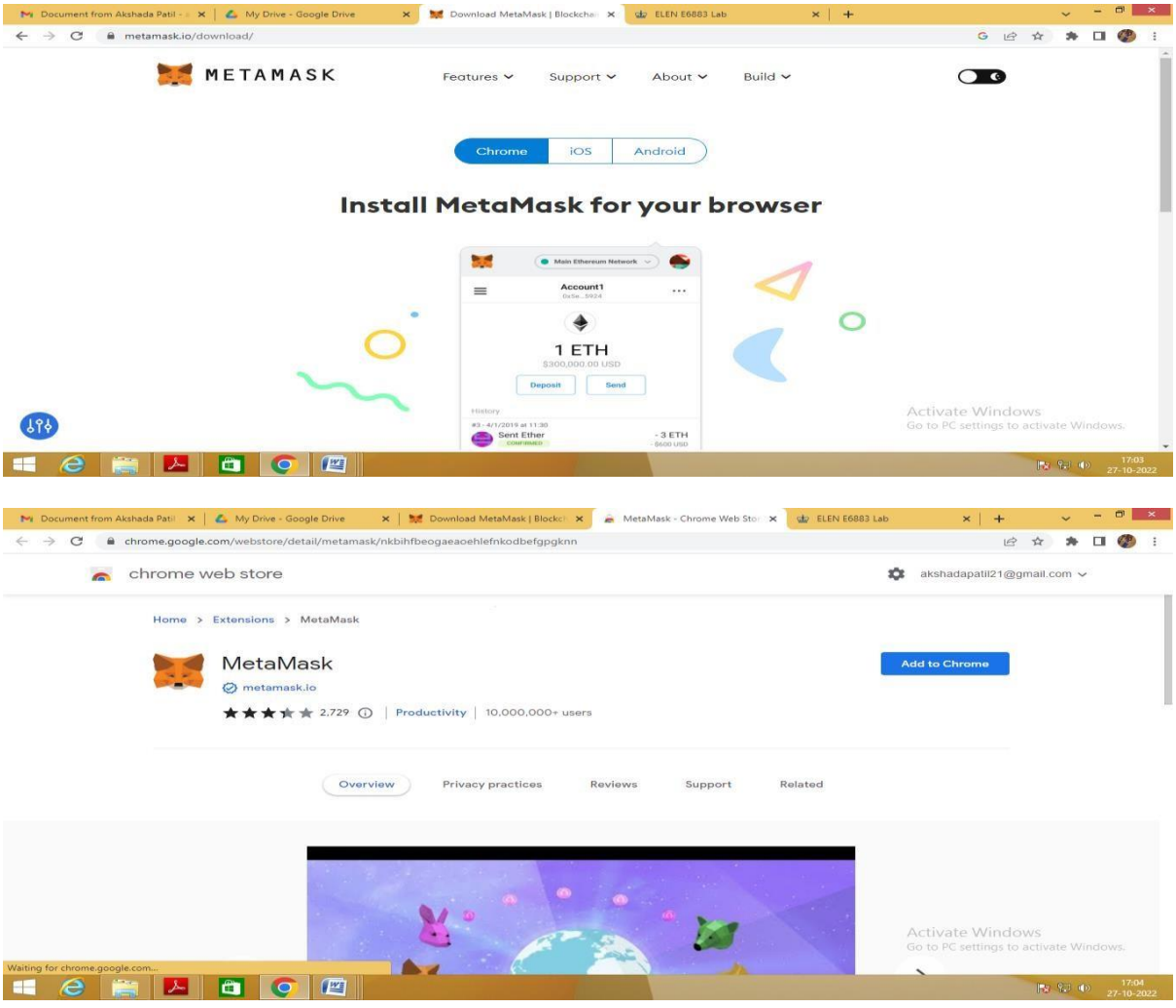
We are now able to use interact with MetaMask

3. Steps to create Metamask:

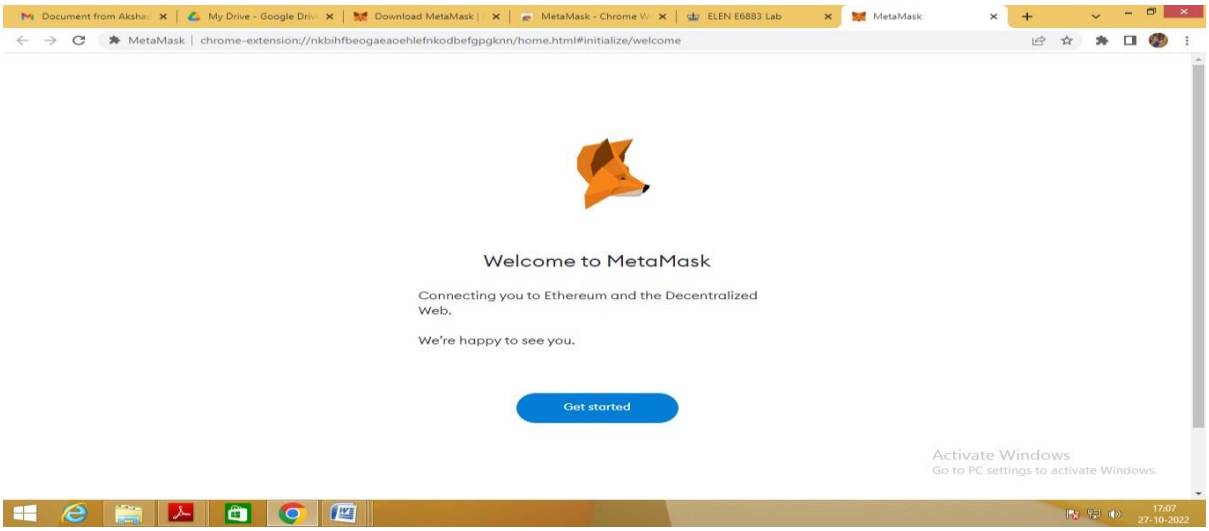
Step 1 : Search on Google the Metamask



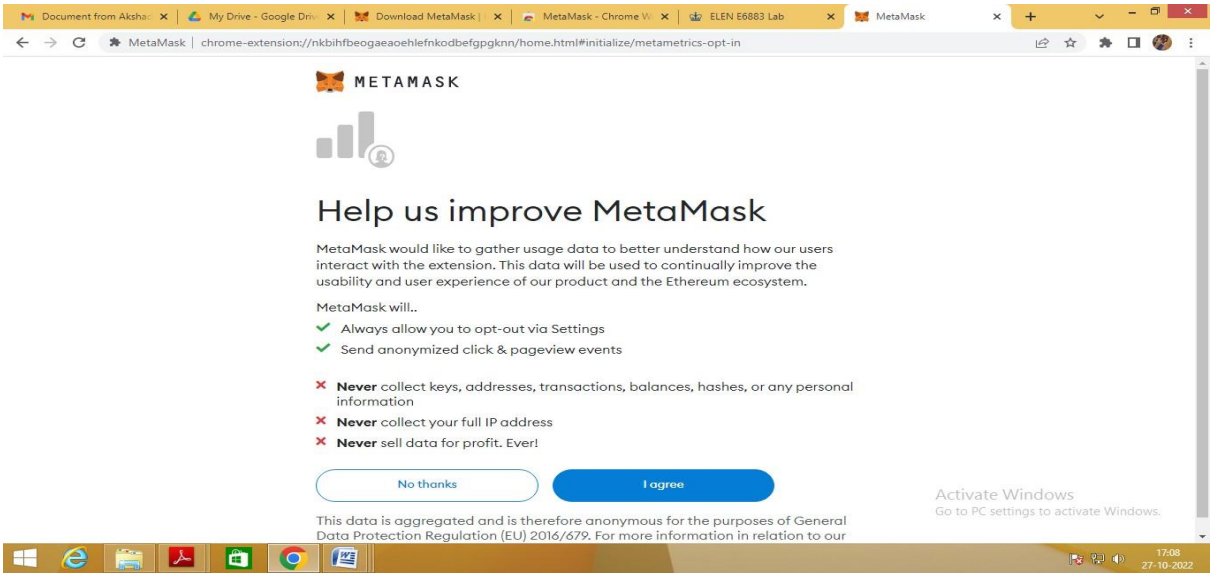
Step 2: Download the Metamask. Click on the button “Add to Chrome”



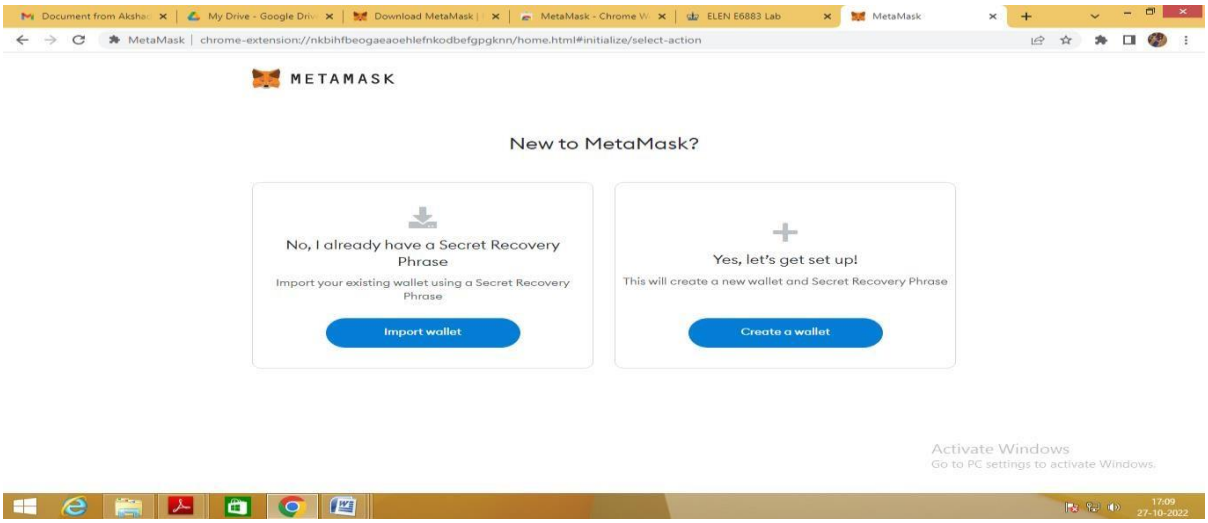
Step 3 : Metamask wallet installation. Click on the Metamask extension and click on “Get Started”



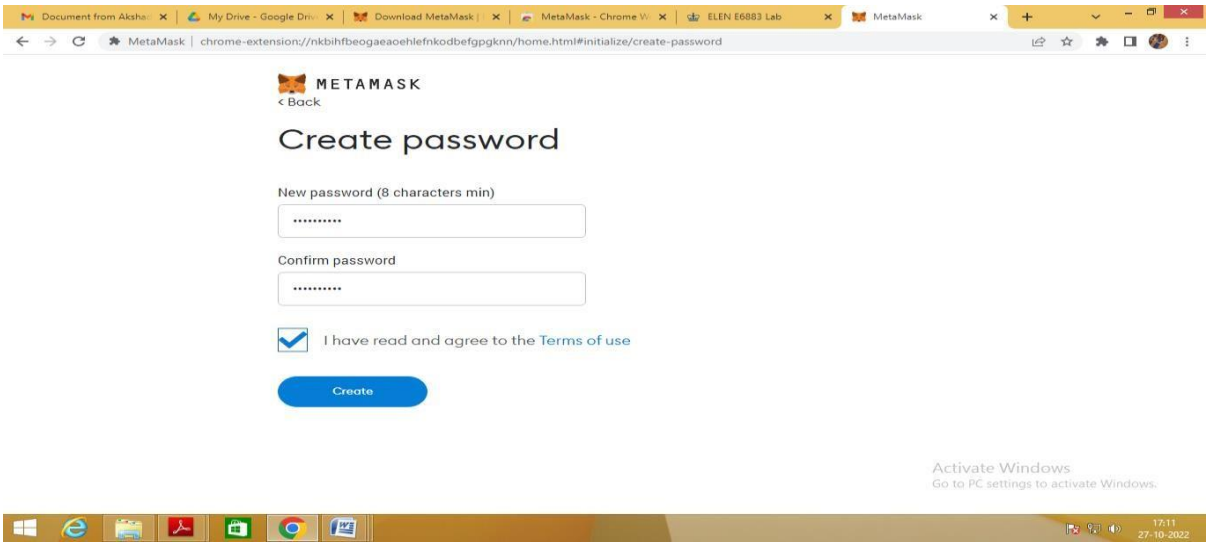
Step 4 : Click on “I Agree”



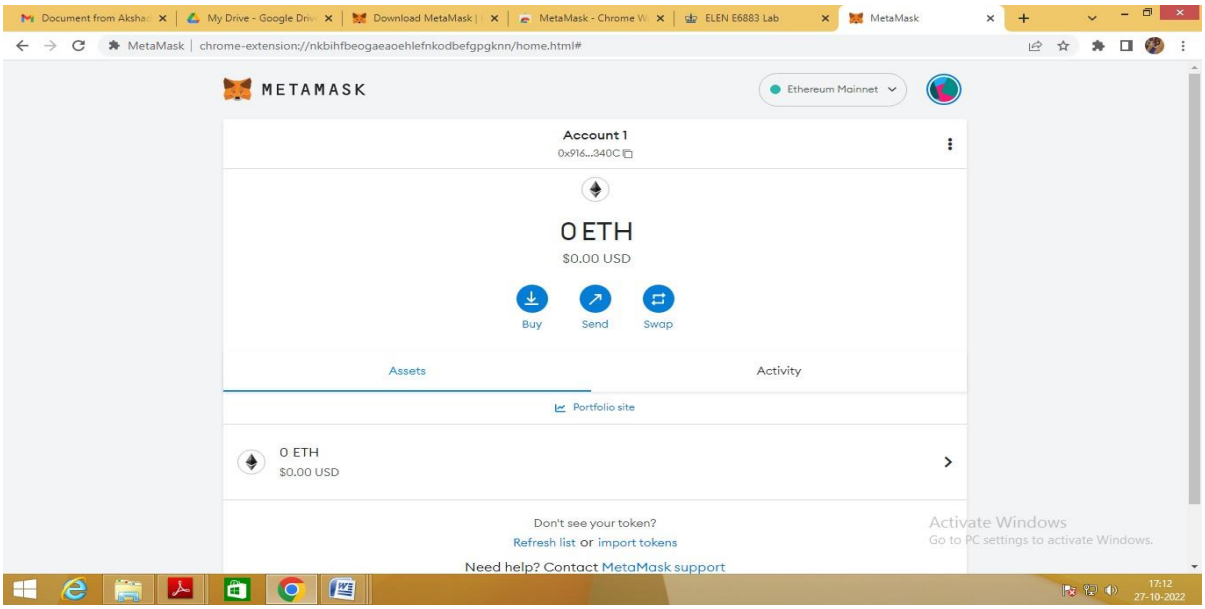
Step 5: Click on the “Create a Wallet”



Step 6: Entre the Password ”********” and confirm password “**********” Tick on I Agree. Click on Create.



Step 7: “Metamask” Account created successfully.



Conclusion: Successfully created Metamask Wallet.

Assignment 3: Write a smart contract on a test network, for bank account of a customer for following operations

1. Deposit money

2. Withdraw money

3. Show balance

Objective: Understand and explore the working of Blockchain Technology and its applications.

Course Outcome:

CO6: Interpret the basic concepts in Blockchain technology and its applications.

Theory :

What Is a Smart Contract?

A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized blockchain network. The code controls the execution, and transactions are trackable and irreversible.

Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism.

While blockchain technology has come to be thought of primarily as the foundation for bitcoin, it has evolved far beyond underpinning the virtual currency.

Anyone can write a smart contract and deploy it to the network. You just need to learn how to code in a smart contract language, and have enough ETH to deploy your contract. Deploying a smart contract is technically a transaction, so you need to pay Gas in the same way you need to pay gas for a simple ETH transfer.

Solidity

Solidity is an object-oriented programming language created specifically by the Ethereum Network team for constructing and designing smart contracts on Blockchain platforms. It's used to create smart contracts that implement business logic and generate a chain of transaction records in the blockchain system.

Solidity is the most commonly used language for writing and deploying smart contracts on the Ethereum chain. It’s a statically typed curly-braces programming language that has familiar features that you might recognize from other languages.

“Solidity looks rather high-level but is still very close to the [Ethereum Virtual Machine],” Christian Reitwiessner, creator of Solidity, said in a 2020 interview. “At the same time,

people with some background in programming usually understand what Solidity code is about."

Solidity's main influences are JavaScript, C++, and Python. If you have a solid (sorry) understanding of those languages, then picking up Solidity is relatively easy.

The Remix Editor

What is a remix platform?

Remix is a powerful, open source tool that helps you write Solidity contracts straight from the browser. Written in JavaScript, Remix supports both usage in the browser and locally. Remix also supports testing, debugging and deploying of smart contracts and much more.

Remix IDE is used for the entire journey of smart contract development by users at every knowledge level. It requires no setup, fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. The IDE comes in 2 flavors (web app or desktop app) and as a VSCode extension.

Remix Online IDE, see: https://remix.ethereum.org

Supported browsers: Firefox, Chrome, Brave. We do not support Remix's use on tablets or mobile devices.

Remix Desktop IDE, see releases: https://github.com/ethereum/remix-desktop/releases

Steps to develop an Ethereum Smart Contract

Step 1: Create a wallet at meta-mask

Step 2: Select any one test network

Step 3: Add some dummy Ethers to your wallet

Step 4: Use editor remix to write the smart contract in Solidity

Step 5: Create a .sol extension file

Step 6: A sample smart contract code to create ERC20 tokens

Step 7: Deploy your contract

Banking Smart Contract

This smart contract will have all basic functionalities like ,

1. Account Creation

2. Deposit Amount

3. Withdraw amount

4. Transfer Amount

5. Send Amount to wallet

First need to add solidity compiler version

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.7.0;
```

Then creating Banking contract ,

```
contract banking{

...

}
```

now let's create variables or objects

```
mapping(address=>uint) public userAccount;

mapping(address=>bool) public userExists;
```

Here userAccount will contain amount for each registered acccount and userExists for account restrictions

Now Create functions for each mentioned operations ,

1. createAcc() functions : Here we create user account using boolean method by making userExists mapping true after using createAcc() function

2. deposit() function : With the help of userExists mapping we are only allowing registered users to deposit into our Smart Contract Bank .

3. withdraw(uint amount) function :

4. TransferAmount() function : Here TransferAmount function transfers amount from one account to other account in the bank only and both users must have created account on the bank to use this function

5. sendAmount() function : Here sender's amount will be transfered from account in the bank to othe receiver's wallet.

Conclusion : In this way we study what is smart contract and how to write and deploy it.

Assignment 4: Write a program in solidity to create student data. Use the following constructs:

1. Structures

2. Arrays

3. Fallback

Deploy this as smart contract on ethereum and observe the transaction fee and Gas values.

Objective: Understand and explore the working of Blockchain Technology and its applications.

Course Outcome:

CO6: Interpret the basic concepts in Blockchain technology and its applications.

Theory :

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state.

Following are the some constructs of solidity:

1. Structures :

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 data type.

Syntax:

```
struct <structure_name> {
  <data type> variable_1;
  <data type> variable_2;
}
```

For accessing any element of the structure, ‘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 the same data types in which each and every element has a specific location called index. Instead of creating numerous individual variables of the same type, we just declare one array of the 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 an Array

To declare an array in Solidity, the data type of the elements and the number of elements should be specified. The size of the array must be a positive integer and data type should be a valid Solidity type

Syntax:

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

Fixed-size Arrays

The size of the array should be predefined. The total number of elements should not exceed the size of the array. If the size of the array is not specified then the array of enough size is created which is enough to hold the initialization.

Dynamic Array:

The size of the 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 the array will be determined

Array Operations

1. Accessing Array Elements: The elements of the array are accessed by using the index. If you want to access ith element then you have to access (i-1)th index.

2. Length of Array: Length of the array is used to check the number of elements present in an array. The size of the memory array is fixed when they are declared, while in case the dynamic array is defined at runtime so for manipulation length is required.

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

4. Pop: Pop is used when the last element of the array is to be removed in any dynamic array.

3. Fallback :

The solidity fallback function is executed if none of the other 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 without

any data. To receive Ether and add it to the total balance of the contract, the fallback function must be marked payable. If no such function exists, the contract cannot receive Ether through regular transactions and will throw an exception.

Properties of a fallback function:

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 for as to make this function call as cheap as possible.

Conclusion : In this way we studied what is smart contract and how to create smart contract for student data using different constructs.

Assignment 5: Write a Survey report on types of blockchains and its real time use cases.

Soln. : Prepare survey report which includes following points:

1. Introduction of Blockchain

2. Overview of Blockchain history

3. Types of Blockchain with real time use cases of each

4. Applications of Blockchain

Conclusion : .....