Department of Computer Science

Sub.:MCA302(Blockchain) **Student Reg.No.:_____** **Year : 2025-26** **Subject Teacher : Dhiren B. Patel**

No	Problem Definition	Assignment Date	Submission Date	Sign of T.A.	Sign of Teacher	Grade	Remark
1	Using https://www.blockchain.com/ show information of Blocks and Transaction. Also search specific Blocks and Transaction. step 1: Open https://www.blockchain.com/ step 2: than go to Explorer menu 2.1 view Blocks 2.1.1 show miner 2.2 view Transactions 2.2.1 show transaction details step 3:Search your transaction and block **[Unit-1]**						
2	Show eth-converter.com and explain it. Install Metamask in you browser and get Eth in that. Install Ganache in windows OS. Show remix.ethereum.org and explain it. Show etherscan.io and explain it. Show andersbrownworth.com/blockchain and explain it. **[Unit-1]**						
3	**Basic Solidity:** • Versioning • Compiling • Contract Declaration • Types & Declaring Variables o uint256, int256, bool, string, address, bytes32 • Default Initializations • Comments • Functions • Deploying a Contract • Calling a public state-changing Function • Visibility • Scope • View & Pure Functions • Structs • Intro to Storage • Arrays - Dynamic & Fixed sized • Compiler Errors and Warnings • Memory • Mappings • SPDX License • Recap **[Unit-2]**						

4	**Storage Factory** Inheritance, Factory Pattern, and Interacting with External Contracts • Factory Pattern • Imports • Deploy a Contract From a Contract • Interact With a Deployed Contract • Recap **[Unit-2]**						
5	**Fund Me** Payable, msg.sender, msg.value, Units of Measure • Payable • Wei/Gwei/Eth Converter • msg.sender&msg.value **[Unit-2]**						

6	**Chainlink Oracles**
	
	• Decentralized Oracle Network Chainlink
	  ○ Blockchains can't make API calls
	  ○ Centralized Nodes are Points of Failure
	• data.chain.link
	• Getting External Data with Chainlink Oracles
	  ○ Chainlink
	  ○ Getting Price Information **[Unit-3]**						

7	**Importing from NPM and Advanced Solidity**
	• Decimals/Floating Point Numbers in Solidity
	• latestRoundData
	• Importing from NPM in Remix
	• Interfaces
	  o Introduction to ABIs
	• Getting Price Feed Addresses
	• getPrice
	• Tuples
	  o Unused Tuple Variables
	• Matching Units (WEI/GWEI/ETH)
	• getConversionRate
	• Matching Units (Continued)
	• SafeMath& Integer Overflow
	  o using keyword
	  o Libraries
	  o SafeMath PSA
	• Setting a Threshold
	• Require
	• Revert
	• Withdraw Function
	• Transfer
	• Balance
	• this
	• Contract Owners
	• Constructor
	• ==
	• Modifiers
	• Resetting
	• for loop
	• Array Length
	• Forcing a Transaction
	• Recap **[Unit-3]**						

8	**Web3.py Simple Storage**
	
	Installing VSCode, Python, and Web3
	
	• Developer Bootcamp Setup Instructions (metamask, vscode, python, nodejs..)
	• VSCode
	• VSCode Crash Course
	• Extensions
	• Short Cuts:
	  o VSCode Shortcuts
	  o VSCodeMacOS Shortcuts
	• Python
	  o Install Troubleshooting
	• Terminal
	• Making a directory/Folder
	• Opening the folder up with VSCode
	• Creating a new file
	• Syntax Highlights
	• Remember to save!
	• Setting linting compile version
	• VSCode Solidity Settings
	  o Formatting & Format on Save
	  o Solidity Prettier
	  o Python Black
	  o Pip **[Unit-3]**						

9	**Our First Python Script with Web3.py - Deploying a Contract**
	• Reading our solidity file
	• Running a Python Script in the Terminal
	• Windows Shortcuts
	• Compiling in Python
	• py-solc-x
	  o compile_standard
	• Colorized Brackets
	• JSON ABI
	• Saving Compiled Code
	• Formatting JSON
	• Deploying in Python
	  i. Get Bytecode
	  ii. Get ABI
	  iii. Choose Blockchain to Deploy To
	  o Local Ganache Chain						

	▪ Ganache UI ▪ Ganache Command Line Web3.py HTTP / RPC Provider Private Keys MUST start with "0x" Contract Object Building a Transaction Account Nonce Calling "Constructor" Transaction Parameters Signing the Transaction NEVER put your private key directly in your code Setting Environment Variables (Windows) Exported Environment Variables Only Last the Duration of the Shell/Terminal Private Key PSA .env file .gitignore Loading .env File in Python o python-dotenv Viewing our Transaction / Deployment in Ganache Waiting for Block Confirmations **[Unit-4]**						
10	**Interacting with Our Contract in Python & Web3.py** • Things you always need i. Contract Address ii. Contract ABI • Getting address from transaction receipt • Calling a view function with web3.py o Call vs Transact Updating State with Web3.py ganache-cli o Installing Ganache ▪ Install Nodejs ▪ Install Yarn Working with ganache-cli Open a new terminal in the same window Deploying to a testnet Infura Alchemy Using Infura RPC URL / HTTP Provider Chain Ids Wow this seems like a lot of work... Is there a better way? **[Unit-4]**						

11	**Brownie Simple Storage**
	
	## Brownie Introduction
	
	- Some Users:
	  - https://yearn.finance/
	  - https://curve.fi/
	  - https://badger.finance/
	
	## Installing Brownie
	
	- Installing Brownie
	  - Install pipx
	  - pipx install eth-brownie
	  - Testing Successful Install
	
	## Brownie Simple Storage Project
	
	- A new Brownie project with `brownie init`
	  - Project Basic Explanation
	- Adding `SimpleStorage.sol` to the `contracts` folder
	- Compiling with `brownie compile`
	- Brownie deploy script
	  - `def main` is brownie's entry point
	- brownie defaults to a `development ganache` chain that it creates
	- Placing functions outside of the `main` function
	- brownie `accounts`
	  - 3 Ways to Add Accounts
	  a. `accounts[0]`: Brownie's "default" ganache accounts
	  ▪ Only works for local ganache
	  b. `accounts.load("...")`: Brownie's encrypted command line (MOST SECURE)
	  ▪ Run `brownie accounts new <name>` and enter your private key and a password
	  c. `accounts.add(config["wallets"]["from_key"])`: Storing Private Keys as an environment variable, and pulling from our `brownie-config.yaml`
	  ▪ You'll need to add `dotenv: .env` to your `brownie-config.yaml` and have						

	a `.env` file
	• Importing a Contract
	• Contract.Deploy
	• View Function Call in Brownie
	• State-Changing Function Call in Brownie / Contract Interaction
	• `transaction.wait(1)`
	
	Testing Basics
	
	• `test_simple_storage.py`
	• Arrange, Act, Assert
	• `assert`
	• `brownie test`
	• `test_updating_storage`
	• Pytest / Brownie Test Tips
	• Deploy to a Testnet
	• `brownie networks list`
	• Development vs Ethereum
	  ○ Development is temporary
	  ○ Ethereum networks persist
	• RPC URL / HTTP Provider in Brownie
	• The network flag
	  ○ `list index out of range`
	• `get_account()`
	• `networks.show_active()`
	• build/deployments
	• Accessing previous deployments
	• Interacting with contracts deployed in our brownie project **[Unit-4]**							

Grade	Marks
A	18 - 20
B	14-17
C	11-13
D	8-10
E	5-7
F	1-4
G	Absent