

## Department of Computer Science

Sub.:MCA302(Blockchain)

Student Reg.No.: \_\_\_\_\_

Year : 2024-25

Subject Teacher : Dhiren B. Patel

No	Problem Definition	Assignment Date	Submission Date	Sign of T.A.	Sign of Teacher	Grade	Remark
1	<p>Using <a href="https://www.blockchain.com/">https://www.blockchain.com/</a> show information of Blocks and Transaction. Also search specific Blocks and Transaction. step 1: Open <a href="https://www.blockchain.com/">https://www.blockchain.com/</a> 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</p>						
2	<p>Show <a href="https://eth-converter.com/">eth-converter.com</a> and explain it. Install Metamask in you browser and get Eth in that. Install Ganache in windows OS. Show <a href="https://remix.ethereum.org">remix.ethereum.org</a> and explain it. Show <a href="https://etherscan.io">etherscan.io</a> and explain it. Show <a href="https://andersbrownworth.com/blockchain">andersbrownworth.com/blockchain</a> and explain it.</p>						
3	<p><b>Basic Solidity:</b></p> <ul style="list-style-type: none"> <li>• Versioning</li> <li>• Compiling</li> <li>• Contract Declaration</li> <li>• Types &amp; Declaring Variables <ul style="list-style-type: none"> <li>◦ uint256, int256, bool, string, address, bytes32</li> </ul> </li> <li>• Default Initializations</li> <li>• Comments</li> <li>• Functions</li> <li>• Deploying a Contract</li> <li>• Calling a public state-changing Function</li> <li>• Visibility</li> <li>• Scope</li> <li>• View &amp; Pure Functions</li> <li>• Structs</li> <li>• Intro to Storage</li> <li>• Arrays - Dynamic &amp; Fixed sized</li> <li>• Compiler Errors and Warnings</li> <li>• Memory</li> <li>• Mappings</li> <li>• SPDX License</li> <li>• Recap</li> </ul>						

4	<b>Storage Factory</b>  Inheritance, Factory Pattern, and Interacting with External Contracts <ul style="list-style-type: none"> <li>• Factory Pattern</li> <li>• Imports</li> <li>• Deploy a Contract From a Contract</li> <li>• Interact With a Deployed Contract</li> <li>• Recap</li> </ul>						
5	<b>Fund Me</b>  Payable, msg.sender, msg.value, Units of Measure <ul style="list-style-type: none"> <li>• Payable</li> <li>• Wei/Gwei/Eth Converter</li> <li>• msg.sender&amp;msg.value</li> </ul>						
6	<b>Chainlink Oracles</b> <ul style="list-style-type: none"> <li>• Decentralized Oracle Network Chainlink <ul style="list-style-type: none"> <li>◦ Blockchains can't make API calls</li> <li>◦ Centralized Nodes are Points of Failure</li> </ul> </li> <li>• data.chain.link</li> <li>• Getting External Data with Chainlink Oracles <ul style="list-style-type: none"> <li>◦ Chainlink</li> <li>◦ Getting Price Information</li> </ul> </li> </ul>						



8	<p><b>Web3.py Simple Storage</b></p> <p><b>Installing VSCode, Python, and Web3</b></p> <ul style="list-style-type: none"> <li>• Developer Bootcamp Setup Instructions (metamask, vscode, python, nodejs..)</li> <li>• VSCode</li> <li>• VSCode Crash Course</li> <li>• Extensions</li> <li>• Short Cuts: <ul style="list-style-type: none"> <li>◦ VSCode Shortcuts</li> <li>◦ VSCodeMacOS Shortcuts</li> </ul> </li> <li>• Python <ul style="list-style-type: none"> <li>◦ Install Troubleshooting</li> </ul> </li> <li>• Terminal</li> <li>• Making a directory/Folder</li> <li>• Opening the folder up with VSCode</li> <li>• Creating a new file</li> <li>• Syntax Highlights</li> <li>• Remember to save!</li> <li>• Setting linting compile version</li> <li>• VSCode Solidity Settings <ul style="list-style-type: none"> <li>◦ Formatting &amp; Format on Save</li> <li>◦ Solidity Prettier</li> <li>◦ Python Black</li> <li>◦ Pip</li> </ul> </li> </ul>					
9	<p><b>Our First Python Script with Web3.py - Deploying a Contract</b></p> <ul style="list-style-type: none"> <li>• Reading our solidity file</li> <li>• Running a Python Script in the Terminal</li> <li>• Windows Shortcuts</li> <li>• Compiling in Python</li> <li>• py-solc-x <ul style="list-style-type: none"> <li>◦ compile_standard</li> </ul> </li> <li>• Colorized Brackets</li> <li>• JSON ABI</li> <li>• Saving Compiled Code</li> <li>• Formatting JSON</li> <li>• Deploying in Python <ol style="list-style-type: none"> <li>Get Bytecode</li> <li>Get ABI</li> <li>Choose Blockchain to Deploy To <ul style="list-style-type: none"> <li>◦ Local Ganache Chain</li> </ul> </li> </ol> </li> </ul>					

	<ul style="list-style-type: none"> <li>▪ Ganache UI</li> <li>▪ Ganache Command Line</li> </ul> <p>Web3.py</p> <p>HTTP / RPC Provider</p> <p>Private Keys MUST start with "0x"</p> <p>Contract Object</p> <p>Building a Transaction</p> <p>Account Nonce</p> <p>Calling "Constructor"</p> <p>Transaction Parameters</p> <p>Signing the Transaction</p> <p>NEVER put your private key directly in your code</p> <p>Setting Environment Variables (Windows)</p> <p>Exported Environment Variables Only Last the Duration of the Shell/Terminal</p> <p>Private Key PSA</p> <p>.env file</p> <p>.gitignore</p> <p>Loading .env File in Python</p> <ul style="list-style-type: none"> <li>○ python-dotenv</li> </ul> <p>Viewing our Transaction / Deployment in Ganache</p> <p>Waiting for Block Confirmations</p>						
10	<p><b>Interacting with Our Contract in Python &amp; Web3.py</b></p> <ul style="list-style-type: none"> <li>• Things you always need             <ol style="list-style-type: none"> <li>i. Contract Address</li> <li>ii. Contract ABI</li> </ol> </li> <li>• Getting address from transaction receipt</li> <li>• Calling a view function with web3.py             <ul style="list-style-type: none"> <li>○ Call vs Transact</li> </ul> </li> </ul> <p>Updating State with Web3.py</p> <p>ganache-cli</p> <ul style="list-style-type: none"> <li>○ Installing Ganache             <ul style="list-style-type: none"> <li>▪ Install Nodejs</li> <li>▪ Install Yarn</li> </ul> </li> </ul> <p>Working with ganache-cli</p> <p>Open a new terminal in the same window</p> <p>Deploying to a testnet</p> <p>Infura</p> <p>Alchemy</p> <p>Using Infura RPC URL / HTTP Provider</p> <p>Chain Ids</p> <p>Wow this seems like a lot of work... Is there a better way?</p>						

11	<div><h2>Brownie Simple Storage</h2><h3>Brownie Introduction</h3><ul style="list-style-type: none"><li>Some Users:<ul style="list-style-type: none"><li><a href="https://yearn.finance/">https://yearn.finance/</a></li><li><a href="https://curve.fi/">https://curve.fi/</a></li><li><a href="https://badger.finance/">https://badger.finance/</a></li></ul></li></ul><h3>Installing Brownie</h3><ul style="list-style-type: none"><li>Installing Brownie<ul style="list-style-type: none"><li>Install pipx</li><li><code>pipx install eth-brownie</code></li><li>Testing Successful Install</li></ul></li></ul><h3>Brownie Simple Storage Project</h3><ul style="list-style-type: none"><li>A new Brownie project with <code>brownie init</code><ul style="list-style-type: none"><li>Project Basic Explanation</li></ul></li><li>Adding <code>SimpleStorage.sol</code> to the <code>contracts</code> folder</li><li>Compiling with <code>brownie compile</code></li><li>Brownie deploy script<ul style="list-style-type: none"><li><code>def main</code> is brownie's entry point</li></ul></li><li>brownie defaults to a development ganache chain that it creates</li><li>Placing functions outside of the <code>main</code> function</li><li>brownie accounts<ul style="list-style-type: none"><li>3 Ways to Add Accounts<ol style="list-style-type: none"><li><code>accounts[0]</code>: Brownie's "default" ganache accounts<ul style="list-style-type: none"><li>Only works for local ganache</li></ul></li><li><code>accounts.load("...")</code>: Brownie's encrypted command line (MOST SECURE)<ul style="list-style-type: none"><li>Run <code>brownie accounts new &lt;name&gt;</code> and enter your private key and a password</li></ul></li><li><code>accounts.add(config["wallets"]["from_key"])</code>: Storing Private Keys as an environment variable, and pulling from our <code>brownie-config.yaml</code><ul style="list-style-type: none"><li>You'll need to add <code>dotenv: .env</code> to your <code>brownie-config.yaml</code> and have</li></ul></li></ol></li></ul></li></ul></div>					
----	--	--	--	--	--	--

	<p>a .env file</p> <ul style="list-style-type: none"> <li>• Importing a Contract</li> <li>• Contract.Deploy</li> <li>• View Function Call in Brownie</li> <li>• State-Changing Function Call in Brownie / Contract Interaction</li> <li>• transaction.wait(1)</li> </ul> <p><b>Testing Basics</b></p> <ul style="list-style-type: none"> <li>• test_simple_storage.py</li> <li>• Arrange, Act, Assert</li> <li>• assert</li> <li>• brownie test</li> <li>• test_updating_storage</li> <li>• Pytest / Brownie Test Tips</li> <li>• Deploy to a Testnet</li> <li>• brownie networks list</li> <li>• Development vs Ethereum <ul style="list-style-type: none"> <li>◦ Development is temporary</li> <li>◦ Ethereum networks persist</li> </ul> </li> <li>• RPC URL / HTTP Provider in Brownie</li> <li>• The network flag <ul style="list-style-type: none"> <li>◦ list index out of range</li> </ul> </li> <li>• get_account()</li> <li>• networks.show_active()</li> <li>• build/deployments</li> <li>• Accessing previous deployments</li> <li>• Interacting with contracts deployed in our brownie project</li> </ul>						

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