CRYSTALIZE.DEV BACKGROUND ASSIGNMENT

Conceptual assessment

CS

1. What is inheritance? What programming paradigm (functional, object oriented, imperative, ...) generally uses inheritance?

Answer:

- Inheritance in programming generally refers to the event where one class (child/sub class) inherits/gains the attributes and methods of another class (parent/superclass).
- Object oriented programming paradigm significantly utilize inheritance as a tool to facilitate the reuse and extension of existing code with out the need to modify it.

2. What is the difference between the stack and the heap in computer memory?

Answer:

- Structure: stack as a linear data structure, differs from heap which is a tree-like structure.
- Storage: stack as a global variable storage usually store the reference to the objects (link) while heap as a dynamic storage contains the objects directly.
- Order: stack are LIFO (Last In First Out) while heap access has no particular order (dynamic memory allocation).

3. What are different data types for variables?

- **Answer:** Different types of data for variables:
    - Integer / Unsigned integer (int/uint)
    - String/Character
    - Boolean
    - Float
    - Long/Short

4. What is a class?

- **Answer:**
    - A class in programming constitutes a framework code for creating objects of a specific kind with defined attributes and methods which is common of every object belonging to that class.

5. What is an object?

Answer:

- An object is an entity/instance belonging to a specific class with corresponding attributes and methods.

6. What are some popular sorting algorithms?

Answer:

- Bubble sort, insertion sort, merge sort, quick sort, etc.

React

1. Can a browser understand JSX? Why / why not?

Answer:

- The browser cannot comprehend JSX as this is not valid JavaScript code (need to use babel to convert jsx into JavaScript and HTML for browser to understand).

2. What are the types of components you can create in react from React's perspective? Name both of them

Answer:

- Types of components: Class and Function components.

3. What is state in React and why is it important?

Answer:

- State is a built-in object used to store property/information of the component.

4. What is a prop?

Answer:

- Prop is used to pass data from one component to another.

5. What is the virtual DOM and why is it important?

Answer:

- Virtual DOM is a virtual representation of the real DOM (Document Object Model). It monitors state and prop changes, helps to improve performance by only updating difference using efficient diff algorithm. It provides an efficient method to update the web application interface.

Git / Git-Hub

1. What is Git and what purpose does it serve? (describe in your own words, in 2-3 sentences)

Answer:

- Git is a distributed software for code management. It is usually utilized in a collective project where multiple developers collaborate on the work so that changes, update in the source code could be shared among them.

2. What is a branch?

Answer:

- A branch is a version of the project repository which demonstrate one of the commits. The primary one – master branch – is the branch that refers to the last commit (every change will be merged into this branch).

3. What is a merge?

Answer:

- Merge is the process of integrating code from one branch into another (e.g. master).

4. What is a fork?

Answer:

- A fork occurs when code from an open source are used to develop new programs/ the same program in a new direction (deviating from the original one).

5. What is the command to go to branch new-feature-1?

Answer:

- git checkout new-feature-1

- git checkout -b new-feature-1 (if the branch does not exist)

6. How do you make a change and push it to branch new-feature-1?

Answer:

```
git add .
git commit -m "changes to new-feature-1"
git push origin new-feature-1
```

7. How do you reflect the changes of the new-feature-1 branch on the master / main branch?

Answer:

```
git checkout master
git merge new-feature-1
```

8. How do you create a pull request on a hosting platform like Git-Hub/Lab?

Answer:

- In the repository site, choose “Branch” -> Choose branch with your commits -> Pull request -> Choose the from and destination branch for merge -> Add title/description -> Create Pull Request

Coding assignment

Please share the GitHub repo with your answers for this section
*(*vivinvinh212/Crystalize_assignment (github.com)*)*

1. What is an array? Create an array with the numbers 1, 2, and 3 in JavaScript

Answer:

- An array constitutes a collection of items (similar items in many languages) stored at contiguous memory locations.

2. Write a simple functional component that implements a counter using the useState hook?
3. Implement a simple sorting algorithm from the ones you fill in question 6. Pseudocode is enough, but functional code is better.
4. Create a simple website that fetches data from any API and display it using React. Examples: weather app, cryptocurrency prices (Coingecko, Coinmarketcap, ... - APIS), ...

Generic questions

1. What is the project that you have built that you are the proudest of?

Answer:

- Although not related to blockchain, I am proudest of my collaborative project on News Aggregator using Text mining. We implemented text mining on information crawl from news site to output recommendations in our self-hosted site for users to read in order to avoid duplicates and gain the best out of their read. What special about this project is that I chose to approach text mining with graph theory instead of the ordinary usage of machine learning (k-clustering, etc.) and surprisingly, the output appears to outperform machine learning considering small dataset and the need for fast processing. This project comprises of many components: web2 functional design, graph theory algorithm development, machine learning comparison implementation, etc.

2. Why would like to learn to build on blockchains?

Answer:

- At first glance, blockchain appears to me as some new and over hyped types of distributed storage. Later on, after discovering about blockchains, distributed ledger, automated smart contracts, I noticed that blockchain has much more to offer than a place to store data. I am into the ideal of no trusted third party (TTP), hence the idea of utilizing the blockchain to create self-functional smart contracts, program to perform a surprising wide range of tasks similar to web2 intrigued me. Hence I decided to put my effort into learning blockchain development to find out what I can build here, different from what can be build on existing frameworks, platforms that allows us to transform the way we operate things on web2: finance, organizations, gaming, etc. This combined with my high curiosity in this newly booming field gives me the courage to give it a try.

3. What are some areas that you are interested in within web3? DeFi, DAOs, public goods, games, infrastructure, security, tooling, ...?

Answer:

- In the web3 field, I am particularly interested in the domain of DeFi as I like the idea of incorporating technology, especially automated system/platform into finance to increase transparency, efficiency and trust from users. Besides that, I am also into security and being a contract auditor due to my inclination to find bugs and fix them.

4. What’s your prior knowledge about blockchains and web3? (bonus)

Answer:

- I have participated in the course on Solidity and JavaScript from FreeCodeCamp and been through some tutorials on Solidity, web3.py and HardHat but have not have a chance to actually learn to code my own application/ take part in a project/bootcamp/intern regarding blockchain development.
- Also, I have some knowledge with interacting with some aspects of the blockchain including coins, NFT, DeFi, DAOs, GameFi, and some X-To-Earn trends through being a crypto/NFT trader: I began my path in 2020, focusing on plays on BSC. ETH, SOL and some other chains where at peak I made around 200$/day with a starting fund of 500$.