**Question 1: What are components in React? Explain the difference between functional components and class components.**

Answer:

Components are the fundamental building blocks of a React application. They are independent, reusable pieces of code that return React elements (JSX) to be rendered to the DOM. Think of them as custom HTML elements that can have their own structure, behavior (logic), and style. The entire React UI is a tree of components nested within each other.

Difference between Functional and Class Components:

| **Feature** | **Functional Components** | **Class Components** |
| --- | --- | --- |
| **Syntax** | A JavaScript function that returns JSX. | A JavaScript class that extends React.Component and must have a render() method which returns JSX. |
| **State Management** | Uses the useState Hook (introduced in React 16.8). | Uses this.state and this.setState(). |
| **Lifecycle Methods** | Uses the useEffect Hook to handle side effects. | Uses dedicated lifecycle methods like componentDidMount, componentDidUpdate, etc. |
| **this Keyword** | Does not use the this keyword. | Requires understanding and binding of the this keyword for methods. |
| **Complexity** | Generally simpler, less code, easier to read and test. | Generally more verbose and complex due to class syntax. |
| **Modern Usage** | The modern standard and recommended way to write components. | Still work but are considered "legacy." New code is typically written with functional compone |

**Question 2: How do you pass data to a component using props?**

Answer:

Props (short for "properties") are the primary way to pass data from a parent component down to a child component. They are read-only, making the data flow predictable and one-directional (unidirectional data flow).

How to pass props:

Define the Data in the Parent: The parent component holds the data (e.g., in its state or a variable).

Pass the Data as an Attribute: In the parent's JSX, you add an attribute to the child component's tag. The attribute name becomes the prop name, and its value becomes the prop value.

Access the Data in the Child: The child component receives all passed data as a single props object (or as the first argument in a functional component). You can then access specific values using dot notation (e.g., props.name).

**Question 3: What is the role of render() in class components?**

Answer:

The render() method is a required and crucial method in every React class component. Its role is singular and vital:

To describe what should be displayed on the screen.

Mandatory: Every class component must have a render() method. It's the only required method.

Returns JSX: The render() method examines this.props and this.state and returns a React element (typically written as JSX) or other valid React nodes (like arrays of elements or strings).

Pure Function: It should be a "pure" function, meaning it does not modify component state, it returns the same output each time it's called with the same inputs (props and state), and it does not directly interact with the browser (e.g., no network requests).

Invoked by React: You don't call render() yourself. React calls it whenever the component needs to be displayed or updated—for example, when its state changes (this.setState()) or when it receives new props from its parent.