**Hands-on: 4. ReactJS-HOL**

**React Component Lifecycle: Understanding Behavior and Control**

In React, components undergo a complete lifecycle, from their creation to their removal from the UI. React offers lifecycle methods (also known as hooks) that empower developers to execute code at specific stages of this lifecycle. Comprehending these methods is crucial for tasks such as data fetching, DOM updates, cleanup, and performance optimization.

**Need and Benefits of the Component Lifecycle**

Need:

React components require a mechanism to handle tasks at different stages, such as when they are mounted, updated, or unmounted. Lifecycle methods provide this necessary control.

**Benefits:**

* Control over behavior during different phases (mount, update, unmount).
* Efficient resource management, including starting and stopping timers.
* Data fetching when a component loads.
* Conditional rendering during updates.
* Cleanup of memory, listeners, or intervals upon component removal.

**Various Lifecycle Hook Methods**

React (Class Component) lifecycle is categorized into three main phases, each with specific methods:

* **Mounting Phase** (Component is being created and inserted into DOM):
  + constructor()
  + static getDerivedStateFromProps()
  + render()
  + componentDidMount()
* **Updating Phase** (Component is re-rendered due to changes in props or state):
  + static getDerivedStateFromProps()
  + shouldComponentUpdate()
  + render()
  + getSnapshotBeforeUpdate()
  + componentDidUpdate()
* **Unmounting Phase** (Component is removed from DOM):
  + componentWillUnmount()
* **Error Handling** (if component throws an error):
  + componentDidCatch()
  + static getDerivedStateFromError()

It is important to note that in Function Components, similar behavior is achieved using React Hooks like

useEffect().

**Sequence of Steps in Rendering a Component**

Here is the typical sequence React follows in Class Components from creation to removal:

* **Mounting Phase** (initial render):
  + constructor()
  + getDerivedStateFromProps()
  + render()
  + componentDidMount()
* **Updating Phase** (on props/state change):
  + getDerivedStateFromProps()
  + shouldComponentUpdate()
  + render()
  + getSnapshotBeforeUpdate()
  + componentDidUpdate()
* **Unmounting Phase:**
  + componentWillUnmount()

Understanding the React component lifecycle is crucial for writing robust, responsive, and efficient applications. Lifecycle methods enable developers to manage component behavior during creation, updates, and removal. With the appropriate use of these hooks, particularly in class components or with

useEffect() in function components, developers can optimize performance, manage side effects, and maintain clean and predictable code.