**4. ReactJS-HOL**

**• Explain the need and Benefits of component life cycle**

The component lifecycle in React refers to the stages a component goes through — from creation (mounting), updates (re-rendering), to removal (unmounting).

Need:

* To control behavior of the component at different phases.
* To fetch data, initialize settings, or clean up resources (like timers or subscriptions).

Benefits:

* Allows executing code at specific points (like after rendering).
* Helps with resource management and performance optimization.
* Makes it easier to manage side effects, API calls, or event listeners.

**• Identify various life cycle hook methods**

React lifecycle methods vary between class components and function components.

Class Component Lifecycle Methods

1. Mounting (when the component is first created):
   * constructor()
   * static getDerivedStateFromProps()
   * render()
   * componentDidMount()
2. Updating (when props or state changes):
   * static getDerivedStateFromProps()
   * shouldComponentUpdate()
   * render()
   * getSnapshotBeforeUpdate()
   * componentDidUpdate()
3. Unmounting (when component is removed from DOM):
   * componentWillUnmount()
4. Error Handling:
   * componentDidCatch()
   * getDerivedStateFromError()

Function Component Lifecycle Hooks

* React provides the useEffect() hook to replicate lifecycle behavior:
  + useEffect(() => { ... }, []) — acts like componentDidMount
  + useEffect(() => { return () => { ... } }) — acts like componentWillUnmount

**• List the sequence of steps in rendering a component**

Rendering sequence for Class Components:

Mounting phase:

scss

constructor() → getDerivedStateFromProps() → render() → componentDidMount()

Updating phase (when props/state change):

scss

getDerivedStateFromProps() → shouldComponentUpdate() → render() → getSnapshotBeforeUpdate() → componentDidUpdate()

Unmounting phase:

scss

componentWillUnmount()