**1. Explain the Need and Benefits of Component Lifecycle**

**Need for Component Lifecycle:**

* React components go through **different stages** (mounting, updating, unmounting).
* Lifecycle methods help us **run code at specific points** during a component's life (e.g., when it's created, updated, or destroyed).
* Useful for:
  + Fetching data from APIs
  + Setting up subscriptions or timers
  + Cleaning up resources before unmounting

**Benefits:**

* **Control over the component's behavior** at each stage.
* **Better performance** by optimizing what updates when.
* **Predictable and organized code**.
* Helps handle **side effects** like API calls, logging, or DOM manipulation.

**2. Identify Various Lifecycle Hook Methods**

**🔸 In Class Components, lifecycle methods are divided by phase:**

**A. Mounting Phase (Component is being created & inserted into the DOM):**

* constructor()
* static getDerivedStateFromProps()
* render()
* componentDidMount()

**B. Updating Phase (Component is being re-rendered due to props/state change):**

* static getDerivedStateFromProps()
* shouldComponentUpdate()
* render()
* getSnapshotBeforeUpdate()
* componentDidUpdate()

**C. Unmounting Phase (Component is being removed from the DOM):**

* componentWillUnmount()

**🔹 In Function Components, lifecycle behavior is handled using Hooks:**

* useEffect(() => {}, []) — Mount
* useEffect(() => {}) — Update
* useEffect(() => { return () => {} }, []) — Unmount
* useLayoutEffect() — Like componentDidMount/componentDidUpdate but synchronous

**3. List the Sequence of Steps in Rendering a Component**

**For Class Components:**

**Mounting (initial render)**

1. constructor()
2. getDerivedStateFromProps()
3. render()
4. componentDidMount()

**Updating (re-render on props/state change)**

1. getDerivedStateFromProps()
2. shouldComponentUpdate()
3. render()
4. getSnapshotBeforeUpdate()
5. componentDidUpdate()

**Unmounting**

1. componentWillUnmount()