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

### **Need:**

The **Component Lifecycle** in React helps developers understand how components behave from creation to removal. It provides built-in methods that allow you to run code at specific points in a component’s life—such as initialization, updating, and destruction.

### **Benefits:**

* Enables control over **component behavior** during its life.
* Helps in **fetching data**, initializing state, and interacting with APIs after component mounts.
* Allows optimization using methods like shouldComponentUpdate().
* Facilitates **cleanup** of resources (e.g., timers, subscriptions) before a component unmounts.
* Useful for **error handling** using lifecycle methods like componentDidCatch().

## **2. Identify Various Lifecycle Hook Methods**

React class components have the following key lifecycle hook methods:

### **Mounting Phase (when the component is being created and inserted into the DOM):**

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

### **Updating Phase (when the component is re-rendered due to state/prop changes):**

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

### **Unmounting Phase (when the component is removed from the DOM):**

* componentWillUnmount()

### **Error Handling:**

* componentDidCatch()
* static getDerivedStateFromError()

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

### **Initial Rendering (Mounting) Sequence:**

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

### **Update Rendering Sequence (on state or props change):**

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

### **Component Removal (Unmounting):**

1. componentWillUnmount()