1. Explain the need and Benefits of component life cycle

React components go through a lifecycle — creation, updating, and destruction. The lifecycle methods/hooks help you:

* Control what happens and when in a component’s life.
* Run code at specific points, like fetching data after rendering, or cleaning up resources before removal.
* Keep the app efficient and predictable.

Benefits of Component life cycle –

* Control over rendering — update UI based on state/props changes.
* Efficient resource management — start/stop things like timers or API calls.
* Debugging ease — know when and why a component re-renders or unmounts.
* Modular logic — separate code for initialization, updates, and cleanup.

1. Identify various life cycle hook methods

|  |  |
| --- | --- |
| constructor() | Initialize state and bind methods |
| componentDidMount() | Run after component is rendered (e.g., API calls) |
| shouldComponentUpdate() | Decide whether to re-render |
| componentDidUpdate() | Run after updates |
| componentWillUnmount() | Cleanup (e.g., timers, subscriptions) |

Hooks:

|  |  |
| --- | --- |
| useEffect() | Acts like componentDidMount, componentDidUpdate, and componentWillUnmount |
| useState() | Manage component’s internal state |
| useRef() | Access DOM or persist values across renders |

1. List the sequence of steps in rendering a component

Component Lifecycle = 3 Phases:

1. Mounting → Setup (e.g., data fetch).
2. Updating → React to state/prop changes.
3. Unmounting → Cleanup

**Outputs –**



