**1. Explain the need and Benefits of component life cycle**

React components go through different stages during their life: when they are **created**, **updated**, and **removed**. This is called the **component life cycle**.

**Why is it needed?**  
It helps us run code at specific times — for example, when a component is first added to the screen or when it receives new data.

**Benefits:**

* You can **fetch data from an API** when the component loads.
* You can **clean up things** like timers when the component is removed.
* Helps in **debugging** by knowing what stage your component is in.
* Improves **performance** by handling logic at the correct time.

**2. Identify various life cycle hook methods**

In class-based components, React gives us some special methods called **life cycle hooks**. These let us do something at each stage.

🔹 Common life cycle methods:

* constructor() – runs when the component is created
* render() – runs when the component shows something on screen
* componentDidMount() – runs once after the component is added to the screen
* componentDidUpdate() – runs when the component updates (like new props or state)
* componentWillUnmount() – runs when the component is removed from the screen
* componentDidCatch() – runs when there’s an error inside the component

**3. List the sequence of steps in rendering a component**

Here is the normal flow when a React class component is rendered:

**- Mounting Phase (component appears on screen for the first time)**

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

**- Updating Phase (when props/state changes)**

1. render()
2. componentDidUpdate()

**- Unmounting Phase (when component is removed)**

1. componentWillUnmount()

**- Error Handling Phase**

1. componentDidCatch()