

Lifecycle Methods (Class Components)

Question 1: What are lifecycle methods in React class components?

Describe the phases of a component's lifecycle.

Ans.

In React class components, lifecycle methods are special methods that run at specific points during a component's life — from creation to removal. They let you control what happens when a component mounts, updates, or unmounts.

Think of a component like a human: born → grows → dies

What are Lifecycle Methods?

Lifecycle methods are predefined methods in class components that React automatically calls at different stages to:

- Initialize state
- Fetch data
- Respond to prop/state changes
- Clean up resources

 These methods are only for class components, not function components (hooks replace them).

Phases of a Component Lifecycle

A React class component has three main lifecycle phases:

1. Mounting Phase (Birth)

When the component is **created and inserted into the DOM**.

Methods in Mounting:

1. constructor()
 - Initialize state
 - Bind methods
2. render()
 - Returns JSX (UI)
 - Pure function (no side effects)
3. componentDidMount()
 - Called after component is rendered to the DOM
 - Best place for:
 - API calls
 - DOM manipulation
 - Subscriptions

2. Updating Phase (Growth)

When the component re-renders due to changes in props or state.

Methods in Updating:

1. render()
 - Re-renders UI
2. componentDidUpdate(prevProps, prevState)
 - Runs after update is committed to DOM
 - Used for:
 - API calls based on prop/state change
 - Conditional updates

3. Unmounting Phase (Death)

When the component is removed from the DOM.

Method in Unmounting:

1. componentWillUnmount()

o Cleanup work:

- Remove event listeners
- Cancel timers
- Unsubscribe from services

Lifecycle Flow (Simple Order)

Mounting:

constructor → render → componentDidMount

Updating:

render → componentDidUpdate

Unmounting:

componentWillUnmount

Question 2: Explain the purpose of componentDidMount(), componentDidUpdate(), and componentWillUnmount().

Ans.

1. componentDidMount()

Purpose

Runs once after the component is mounted (inserted into the DOM).

Why it is used

- Fetch data from APIs
- Set up subscriptions
- Add event listeners
- DOM manipulation

Key Point

- ✓ Runs only one time
- ✓ Safe place for side effects

2. componentDidUpdate(prevProps, prevState)

Purpose

Runs after every update caused by props or state changes.

Why it is used

- Respond to prop/state changes
- Re-fetch data when values change
- Perform actions after DOM update

Important Rule 

Always compare previous values, otherwise it may cause an infinite loop.

3. componentWillUnmount()

Purpose

Runs just before the component is removed from the DOM.

Why it is used

- Cleanup tasks
- Remove event listeners
- Clear timers
- Cancel subscriptions or API calls

Quick Comparison Table :

Method	When It Runs	Main Use
componentDidMount	After first render	API calls, setup
componentDidUpdate	After every update	Respond to changes
componentWillUnmount	Before removal	Cleanup
