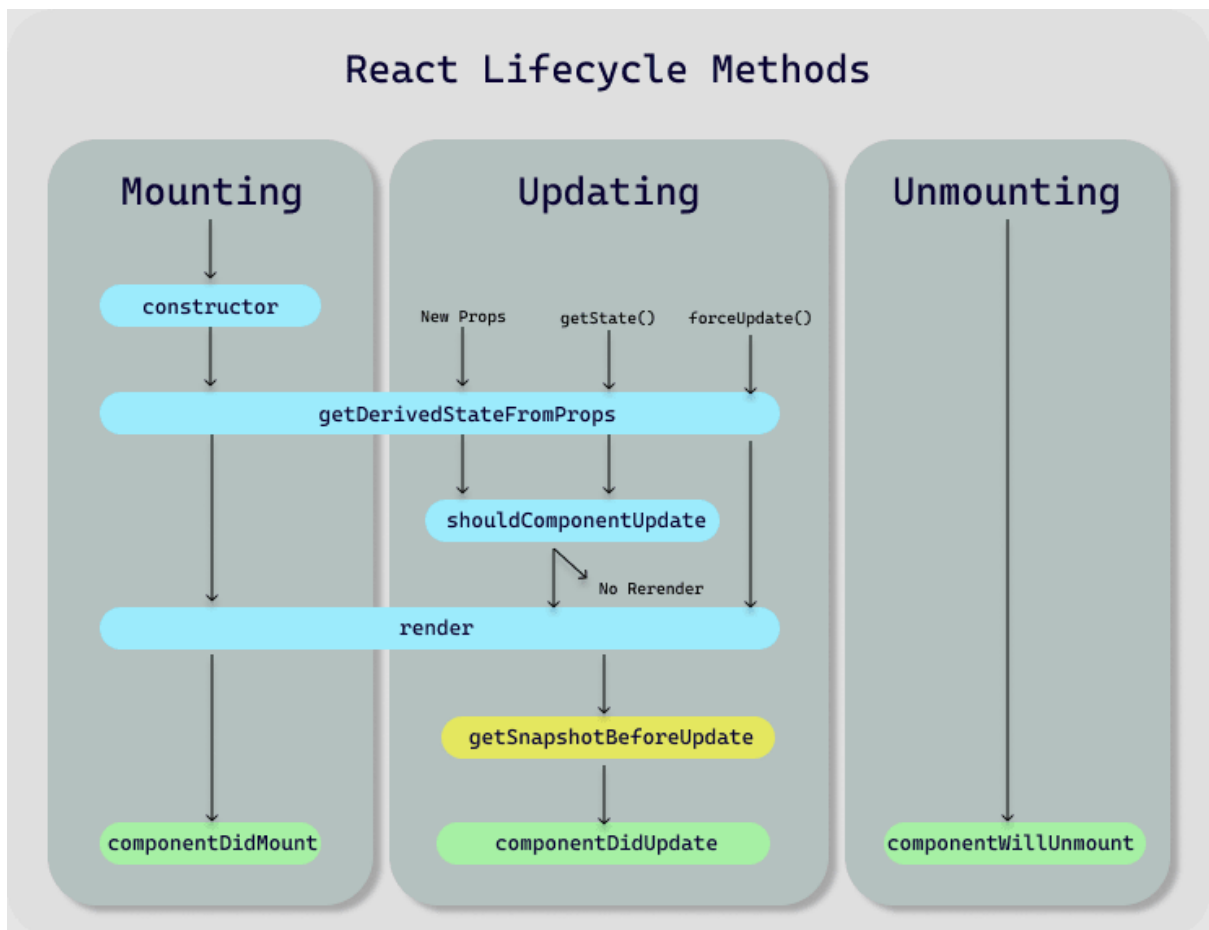


MODULE: 10 List and Hooks

Que - Explain Life cycle in Class Component and functional component with Hooks.

Sol -

In React, both Class Components and Functional Components with Hooks have a life cycle that represents the stages a component goes through from initialization to removal. However, the life cycle methods differ slightly between the two types of components.



Class Component Life Cycle:

1. Initialization: `constructor(props)` method is called, which sets the initial state and binds the `this` keyword.
2. Mounting: `componentWillMount()` method is called before the component is rendered. Then, `render()` method is called to generate the JSX that represents the component, followed by `componentDidMount()` method, which is called after the component is rendered and inserted into the DOM.
3. Updating: `componentWillReceiveProps(nextProps)` method is called when the component is about to receive new props. If the component's state or props change, `shouldComponentUpdate(nextProps, nextState)` method is called to determine whether the component should update. If it returns `true`, then `componentWillUpdate(nextProps, nextState)` method is called before the component is re-rendered. After re-rendering, `componentDidUpdate(prevProps, prevState)` method is called.
4. Unmounting: `componentWillUnmount()` method is called before the component is removed from the DOM.

Functional Component Life Cycle with Hooks:

With the introduction of Hooks in functional components can also manage state and have a life cycle that replaces the need for Class components. The life cycle is similar, but implemented differently:

1. Initialization: The component function is called, and the initial state is set using the `useState()` Hook.
2. Mounting: The `useEffect()` Hook with an empty dependency array is used to replace `componentDidMount()`, which runs after the component is mounted. It can also return a cleanup function that runs when the component is unmounted.
3. Updating: The `useEffect()` Hook with a dependency array is used to replace the `shouldComponentUpdate()` and `componentDidUpdate()` methods. It is called after every re-render, and it can conditionally update the component or run side effects based on the dependency array.

4. Unmounting: The `useEffect()` Hook with a cleanup function is used to replace the `componentWillUnmount()` method. It runs before the component is unmounted.

In summary, Class Components and Functional Components with Hooks both have a similar life cycle that represents the stages a component goes through from initialization to removal. However, the life cycle methods are implemented differently between the two types of components.