Explain Life cycle in Class Component and functional component with Hooks ?

**Class Component Life Cycle:**

1. **Mounting Phase:**
   * **constructor()**: Called when an instance of the component is created. Used for initializing state and binding methods.
   * **static getDerivedStateFromProps()**: Invoked right before rendering when new props or state are received. Rarely used.
   * **render()**: Responsible for rendering the component's UI.
   * **componentDidMount()**: Invoked immediately after the component is inserted into the DOM. Ideal for fetching data or setting up subscriptions.
2. **Updating Phase:**
   * **static getDerivedStateFromProps()**: Similar to mounting phase, but used when the component is re-rendered.
   * **shouldComponentUpdate()**: Determines if the component should re-render. Provides a performance optimization by preventing unnecessary renders.
   * **render()**: Renders the updated UI.
   * **componentDidUpdate()**: Called after the component is re-rendered. Useful for performing side effects, such as updating the DOM in response to prop or state changes.
3. **Unmounting Phase:**
   * **componentWillUnmount()**: Called just before the component is removed from the DOM. Used for cleanup, such as cancelling network requests or clearing up subscriptions.

**Functional Component with Hooks:**

1. **Mounting Phase:**
   * **useState()**: Hook for adding state to functional components.
   * **useEffect(() => {}, [])**: Combines **componentDidMount** and **componentDidUpdate**. The function passed to **useEffect** runs after the component renders. The empty dependency array (**[]**) ensures it only runs once after the initial render.

2.Updating Phase:

useEffect(() => {}): Use this hook without a dependency array to run code on every render.

3.Unmounting Phase:

useEffect(() => () => {}): The cleanup function runs when the component is unmounted.