

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

Ans:

=> In React, components have a lifecycle that describes the phases they go through from birth (mounting) to death (unmounting).

These lifecycles allow you to run code at specific times during a component's existence, such as when it is first rendered, updated, or removed from the DOM.

Lifecycle in Class Components:

* Mounting Phase:

constructor(): This method is called before the component is mounted. It's used for initializing state and binding event handlers.

componentDidMount(): This method is called after the component has been mounted to the DOM.

*Updating Phase:

This method is called after the component has been updated.

*Unmounting Phase:

componentWillUnmount(): This method is called just before the component is unmounted and destroyed.

Lifecycle in Functional Components with Hooks:

Functional components, introduced in React 16.8, can also have lifecycle-like behavior using React Hooks. Here's how you can achieve similar functionality using hooks:

***Mounting and Updating Phase:**

`useEffect()`: This hook allows you to perform side effects in function components. It runs after every render by default, including the initial render.

while class components have specific lifecycle methods, functional components with hooks primarily use the `useEffect` hook to achieve similar functionality for handling mounting, updating, and unmounting phases. Both approaches allow you to manage side effects and perform cleanup in React components.