Module:4(List and Hooks)

**1. Explain Life cycle in Class Component and functional component with Hooks**

Ans. Every Component in React are created (Mounted on the DOM), change by updating and then removed (Unmounted on DOM). This is mentioned as a component lifecycle. There are three methods in React that defines three phases of components life:-

**1. Mounting:-** In this phase our react component mounts on the DOM. And there are two methods to do it

**1. componentWillMount():-** This method is called before the component mounts on DOM or we can say before render method is called.

**2. componentDidMount():-** This methode is called after the component mounted on DOM or we can say after render method is called.

**2. Updating:-** In this phase our component is already created and mounted on DOM. This phase executes when component’s state or props are changes or re-rendering take place. And this will execute by User when they will interact with the component (pressing button ,inputs). There are three methods to do it

**1. componentShouldUpdate():-** This method is called when you want to re-render the component on some condition. By default, it returns true.

**2. componentWillUpdate():-** This method is called before rerendering has done and when you want to do some functionalities just before re-rendering take place.

**3. componentDidUpdate():-**This method is called after the rerendering has done or we can say after component mounted on DOM.

**3. Unmounting:-** Last phase of component lifecycle. In this phase our component is removed or unmounted from DOM. Only one method is there

**1. componentWillUnmount():-** This method is called before the component Unmounting on DOM. Denoting that it will Unmounted soon or we can say end of the component life cycle. Example:- Function Component

import React,{useEffect} from "react";

**Mounting:-**

useEffect(()=>{

console.log("DidMount")

},[])

**Unmountiing:-**

useEffect(()=>{

console.log("DidMount")

return () => console.log("WillUnmount")

},[])

**Updating:-**

useEffect(()=>{

return () => console.log("DidUpdate")

},[data.number])

Example:- Class Component

**Mounting:-**

componentDidMount(){

console.log("DidMount")

}

**Unmounting:-**

componentWillUnmount(){

console.log("WillUnmount")

}

**Updating:-**

componentDidUpdate(){

console.log("DidUpdate”)