***MODULE: 10 List and Hooks***

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

**Ans:-**

* **Class Component:-**
* Each component in React has a lifecycle which you can monitor and manipulate during its three main phases.
* The three phases are:
* componentDidMount() Mounting, Birth component is rendered.
* componentDidUpdate() Updating, update component is updated in the DOM.
* componentWillUnmount() Unmounting, death removed from the DOM.
* **Example:-**

**import React, { Component } from 'react'**

**import Class\_lifeImg from './Class\_lifeImg';**

**export class Class\_Lifecycle extends Component {**

**constructor(){**

**super();**

**this.state={**

**name:"Rajesh Nagar",**

**IsImage:true**

**}**

**}**

**componentDidUpdate(){**

**console.log(' componentDidUpdate / Update')**

**}**

**render() {**

**return (**

**<div className='container mt-5'>**

**<button onClick={()=>this.setState({IsImage:false})}>Hide</button>**

**<button onClick={()=>this.setState({IsImage:true})}>Show</button>**

**{this.state.IsImage ? <Class\_lifeImg/> : null}**

**</div>**

**)**

**}**

**}**

**export default Class\_Lifecycle**

* **Functional Component:-**
* The useEffect Hook allows you to perform side effects in your components.
* Some examples of side effects are: fetching data, directly updating the DOM, and timers.
* useEffect accepts two arguments. The second argument is optional.
* useEffect(<function>, <dependency>)
* useEffect runs on every render.
* That means that when the count changes, a render happens, which then triggers another effect.
* **Example:-**

**import React, { useEffect, useState } from 'react'**

**import Func\_lifeImg from './Func\_lifeImg'**

**function Func\_Lifecycle() {**

**const [data,setData]=useState({**

**name: "Rajesh Nagar",**

**IsImage: true,**

**number:1,**

**})**

**useEffect(()=>{**

**console.log('Main componentDidMount / componentDidUpdate');**

**},[data.number]);**

**return (**

**<div className='container mt-5'>**

**<button onClick={() => setData({...data, number: data.number+1 })}>+</button>**

**<h1>{data.number}</h1>**

**<button onClick={() => setData({...data, number: data.number-1 })}>-</button>**

**<hr />**

**<button onClick={() => setData({...data, IsImage: false })}>Hide</button>**

**<button onClick={() => setData({...data, IsImage: true })}>Show</button>**

**{data.IsImage? <Func\_lifeImg/> : null}**

**</div>**

**)**

**}**

**export default Func\_Lifecycle**