**List and Hooks**

**1.Explain Life cycle in class components and functional components with Hooks.**

**Ans.** there are two types of components - Class Components and Functional Components.

Class Components have a lifecycle with different methods that get called at specific points to facilitate managing state, performing side effects, etc. Some key lifecycle methods include componentDidMount (called after first render), componentDidUpdate (called after re-render), and componentWillUnmount (called before component is destroyed).

Functional Components, on the other hand, historically did not have lifecycle methods or state management capabilities. However, with the introduction of Hooks in React 16.8, functional components gained the ability to manage state and use lifecycle-like capabilities through Hooks such as useState, useEffect, useContext, etc.

The useEffect Hook essentially combines the lifecycle methods like componentDidMount, componentDidUpdate, and componentWillUnmount into a single API, allowing functional components to perform side effects based on the rendered state. Developers can control when side effects run by providing dependencies to useEffect.

So in summary, while Class Components have defined lifecycle methods, Functional Components with Hooks like useEffect provide a way to simulate and manage lifecycle behaviors in a more flexible and reusable manner through function composition.