React Component Lifecycle - Overview

## Explain the need and Benefits of component life cycle

The React component lifecycle allows developers to manage and control what happens at different phases of a component's existence—mounting, updating, and unmounting. It provides hooks (methods) that enable developers to:  
- Initialize data (e.g., fetching from APIs)  
- Update UI based on state/prop changes  
- Clean up before removing a component (e.g., cancel timers or API calls)  
Benefits include better performance, clean structure, efficient resource usage, and improved debugging.

## Identify various life cycle hook methods

The main lifecycle methods (in class components) are:  
1. constructor() – for initializing state.  
2. componentDidMount() – invoked after component is mounted.  
3. shouldComponentUpdate() – determines if a re-render is necessary.  
4. componentDidUpdate() – called after the component updates.  
5. componentWillUnmount() – used for cleanup before the component is destroyed.

## List the sequence of steps in rendering a component

The typical sequence of steps in rendering a React class component is:  
1. constructor()  
2. render()  
3. componentDidMount()  
Then on update:  
4. shouldComponentUpdate()  
5. render()  
6. componentDidUpdate()  
And on removal:  
7. componentWillUnmount()