**Objectives**

* Explain the need and Benefits of component life cycle

In React, the component lifecycle provides a way to manage and control what happens during the various phases of a component’s existence—creation, update, and removal. The **need** for lifecycle methods arises when developers want to perform specific tasks at different points in a component’s journey. For example, fetching data from an API once the component is rendered, or cleaning up timers or event listeners before the component is removed from the DOM. The **benefits** of using lifecycle methods include improved performance, clean resource management, and better control over rendering behavior. They help make components more predictable, maintainable, and efficient by providing hooks into key points of execution.

* Identify various life cycle hook methods

React class components provide several lifecycle hook methods, each serving a purpose during different phases of a component's life. During the mounting phase, the methods are:

* constructor() – for initializing state and binding event handlers,
* static getDerivedStateFromProps() – for updating state based on props,
* render() – to return JSX for rendering the UI, and
* componentDidMount() – for running code after the component is added to the DOM (commonly used for API calls).
* List the sequence of steps in rendering a component

The rendering process of a React component follows a specific sequence of lifecycle methods. During the **mounting phase**, the order is:

* constructor() – initializes the component’s state and bindings.
* static getDerivedStateFromProps() – optionally updates state based on props.
* render() – returns the JSX to display.
* componentDidMount() – executes after the component has been added to the DOM.

Create a new react application using *create-react-app* tool with the name as “blogapp”

**OUTPUT:**

