**Objectives**

* Explain the need and Benefits of component life cycle

Knowing the component lifecycle in React is crucial when dealing with the behavior of components over time. Every component in React has a lifecycle that includes stages such as mounting, updating, and unmounting. React has lifecycle methods (or hooks) that enable developers to execute code at certain places within this cycle—a case in point being loading data after mounting a component or disposing of resources before it gets eliminated from the DOM.

The requirement for lifecycle methods comes into play when a developer wishes to execute activities such as fetching data, setting timers, or DOM manipulation at specific points within the life of a component. This guarantees optimal performance and resource handling. Lifecycle methods bring structure and predictability to the way in which a component will react across its life.

* Identify various life cycle hook methods

In class components, lifecycle hook functions are `constructor()`, `componentDidMount()`, `componentDidUpdate()`, and `componentWillUnmount()`. In functional components, there are hooks such as `useEffect()` that perform similar functions. The common order of steps involved in a component's rendering are: (1) Initialization through constructor, (2) Mounting through `render()` and `componentDidMount()`, (3) Updating through `setState()` or new props invoking `componentDidUpdate()`, and (4) Unmounting through `componentWillUnmount()`.

* List the sequence of steps in rendering a component

React is the waiter who puts in requests from customers and brings them their orders. This process of requesting and serving UI has three steps:

Triggering a render (delivering the guest’s order to the kitchen)

Rendering the component (preparing the order in the kitchen)

Committing to the DOM (placing the order on the table)

