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

Answer: There are two types of component to generate the UI representation of the components.

i. Class Components,

ii. Function Component.

**Life cycle in Class Component**

Here are three phases of class component:

(A)Mounting, (B)Updating, (C)Unmounting.

**(A)Mounting**

(1) constructor(): This method call when a components is being created and initialized.

(2) componentWillMount(): Applied just before rendering for the first time.

(3) render(): This method display the component on web page.

(4) componentDidMount(): Applied after the components are render the DOM.

**(B)Updating**

(1) shouldComponentUpdate(): Decides if the component should update or not based on certain conditions.

(2) componentsWillUpdate(): Applied just before rendering when props or state are being received.

(3) render(): This method display the component on web page.

(4) componentDidUpdate(): Applied after the components are updated in the DOM.

**(C)Unmounting**

(1) co,ponentWillUnmount(): Apply immediately before a component is unmounted and destroyed.

**Here are some example of “Life Cycle in Class Component”**

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

const ExampleFunctionalComponent = () => {

  const [state, setState] = useState(initialState);

  useEffect(() => {

    return () => {

    };

  }, [dependencies]);

  return (

    // Render the component

    <div>

      Example Functional Component

    </div>

  );

}

export default ExampleFunctionalComponent;

**Now we learn about life cycle in function component:**

**Life cycle in Function Component**

Here are three phases of function component:

(A)Mounting, (B)Updating, (C)Unmounting.

**(A) Mounting**

(1) useState(): Initialized state.

(2) useEffect(): Equals to “componentDidMount()” and “componentDidUpdate()” but apply after rendering and re-rendering the component.

(3) useLayoutEffect(): Similar to “useEffect()” but runs synchronously after DOM changes.

**(B) Updating**

(1) useState(): Update state.

(2) useEffect(): Equals to “componentDidUpdate()” apply after re-rendering the component.

(3) useLayoutEffect(): Similar to “useEffect()” but runs synchronously after DOM changes

**(C) Unmounting**

(1) useEffect(): Equals to “componentWillUnmounting()” method but apply before a component unmounted.

**Here are some example of “Life Cycle in Function Component with Hooks”**

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

const ExampleFunctionalComponent = () => {

  const [state, setState] = useState(initialState);

  useEffect(() => {

    return () => {

    };

  }, [dependencies]);

  return (

    // Render the component

    <div>

      Example Functional Component

    </div>

  );

}

export default ExampleFunctionalComponent;