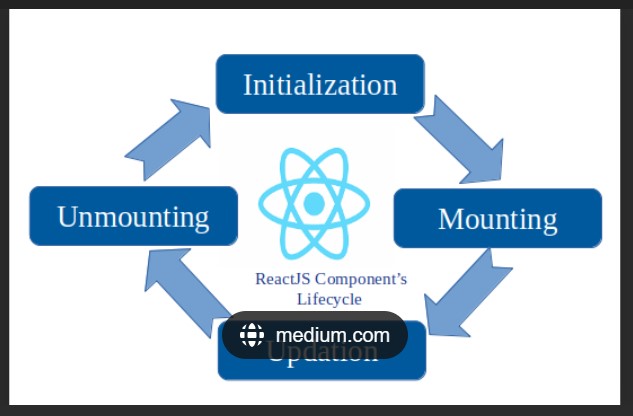
(Q) Explain Life cycle in class Component and Functional Components with Hooks.

Life cycle-



1. Mounting-

[In React, **Mounting** is the initial phase in which the instance of the component is created and inserted into the DOM 1](https://www.w3schools.com/react/react_lifecycle.asp)[2](https://www.geeksforgeeks.org/what-does-it-mean-for-a-component-to-be-mounted-in-reactjs/)[3](https://stackoverflow.com/questions/31556450/what-is-mounting-in-react-js). During this phase, React has four built-in methods that get called in this order:

1. Updating-

The updating phase occurs when a component’s props or state changes.

1. Unmounting-

The unmounting phase occurs when a components is removed from the DOM.

Example-

import React from "react";

import ReactDOM from "react-dom/client";

class Test extends React.Component {

    constructor(props) {

        super(props);

        this.state = { hello: "World!" };

    }

    componentDidMount() {

        console.log("componentDidMount()");

    }

    changeState() {

        this.setState({ hello: "Tops!" });

    }

    render() {

        return (

            <div>

                <h1>

                    Tech

                    {this.state.hello}

                </h1>

                <h2>

                    <a

                        onClick={this.changeState.bind(

                            this

                        )}

                    >

                        Press Here!

                    </a>

                </h2>

            </div>

        );

    }

    shouldComponentUpdate(nextProps, nextState) {

        console.log("shouldComponentUpdate()");

        return true;

    }

    componentDidUpdate() {

        console.log("componentDidUpdate()");

    }

}

const root = ReactDOM.createRoot(

    document.getElementById("root")

);

root.render(<Test />);

Functional components-

Functional Components are some of the more common components that will come across while working in React. These are simply JavaScript functions. We can create a functional components to React by writing a JavaScript function.

Example of functional components-

import React,{useState} from "react";

const FunctionalComponents = ()=>{

    const [count, setCount] = useState(0);

    const increase = () => {

        setCount(count + 1);

    }

    return(

        <div style={{margin:'50px'}}>

            <h3> counter app using Functional Components:</h3>

            <h2>{count}</h2>

            <button onClick={increase}>Add</button>

        </div>

    )

}

export default FunctionalComponents;

Class Component-

This is the bread and butter of most modern web aps built in ReactJS. These components are simple classes (made up of multiple functions that add functionality to the application).

Example of class components.

import React,{Component} from "react";

class ClassComponent extends React.Component{

  constructor() {

    super();

    this.state = {

      count: 0

    };

    this.increase = this.increase.bind(this);

  }

  increase() {

    this.setState({ count: this.state.count +1});

  }

  render () {

    return (

      <div style={{ margin: '50px'}}>

        <h3>counter App Class Component :</h3>

        <h2> {this.state.count}</h2>

        <button onClick={this.increase}>Add</button>

      </div>

    )

  }

}

export default ClassComponents;