**2. React HOL**

**1. Explain React Components**

A **React component** is a **reusable piece of UI** that can be reused and combined to build complex user interfaces.

Components are the **building blocks** of React applications. Each component can be:

* A function or class
* Accept **props** (inputs)
* Manage **state** (optional)
* Return **JSX** to render UI

**2. Differences Between Components and JavaScript Functions**

| **Aspect** | **React Component** | **JavaScript Function** |
| --- | --- | --- |
| Purpose | Builds part of the UI | Performs logic or computation |
| Returns | JSX (UI code) | Any value (number, string, etc.) |
| Lifecycle methods | Yes (for class components) | No |
| State management | Yes (state, hooks) | No |
| Props | Used to pass data into component | Not a native concept |

**3. Types of React Components**

| **Type** | **Syntax Style** | **Supports State?** | **Uses this?** |
| --- | --- | --- | --- |
| **Class Component** | ES6 class | Yes | Yes |
| **Function Component** | ES6 function | Yes (with Hooks) | No |

**4. Explain Class Component**

A **Class Component** is a React component defined using a JavaScript class, extending React.Component.

**Example:**

import React, { Component } from 'react';

class Welcome extends Component {

render() {

return <h1>Hello, {this.props.name}</h1>;

}

}

**Key Features:**

* Uses render() method to return JSX
* Access this.props and this.state
* Supports lifecycle methods (componentDidMount, etc.)

**5. Explain Function Component**

A **Function Component** is a simpler component declared using a **JavaScript function**.

**Example:**

function Welcome(props) {

return <h1>Hello, {props.name}</h1>;

}

Or with Arrow Function:

const Welcome = ({ name }) => <h1>Hello, {name}</h1>;

**Key Features:**

* Simpler syntax
* Uses **Hooks** (useState, useEffect) for state/lifecycle
* No this keyword

**6. Define Component Constructor**

The **constructor** in a class component is a special method used to:

* Initialize state
* Bind methods
* Access props

**Example:**

class MyComponent extends React.Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

}

super(props) is required to access this.props in the constructor.

**7. Define render() Function**

The render() method is required in every **class component**. It **returns the JSX** that React will render to the UI.

**Example:**

class Welcome extends React.Component {

render() {

return <h1>Welcome, {this.props.name}</h1>;

}

}

* It must return **a single JSX element** (wrap multiple in a parent tag or <>...</>)
* Called every time the component updates