**3. ReactJS-HOL**

**1. Explain React Components**

A **React Component** is the fundamental building block of a React application. Components allow developers to **split the UI into independent, reusable pieces**, making it easier to build and maintain complex interfaces.

Each component:

* Can accept **input (props)**
* Can manage its own **internal state**
* Must return **JSX**, which describes what should appear on the screen

React components help in developing scalable and maintainable applications by promoting reusability and modular architecture.

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

| **Feature** | **React Component** | **JavaScript Function** |
| --- | --- | --- |
| Purpose | Builds and renders UI | Performs a specific logic or calculation |
| Return Type | Returns JSX (React elements) | Returns a value (string, number, object, etc.) |
| Integration with React | Directly used in JSX to render elements | Cannot be used directly in JSX without conversion |
| State Management | Can use state (via useState or this.state) | Cannot manage state |
| Lifecycle Methods | Available in class components (e.g., componentDidMount) | Not available |

**3. Identify the Types of Components**

React provides two main types of components:

1. **Class Components**
   * Defined using ES6 class syntax
   * Can use lifecycle methods and maintain internal state
   * Have a render() method
2. **Function Components**
   * Defined using standard JavaScript functions
   * Initially used for UI only, but now support state and side-effects via **hooks**

**4. Explain Class Component**

A **Class Component** is a component defined as a class that extends React.Component. It must include a render() method that returns JSX.

**Example:**

jsx

CopyEdit

import React, { Component } from 'react';

class Welcome extends Component {

render() {

return <h1>Hello from Class Component</h1>;

}

}

Key characteristics:

* Can hold local state using this.state
* Access props via this.props
* Can use lifecycle methods like componentDidMount(), shouldComponentUpdate()

**5. Explain Function Component**

A **Function Component** is a simpler way to write components using JavaScript functions. It can return JSX and with React hooks, it can manage state and side effects.

**Example:**

jsx

CopyEdit

import React from 'react';

function Welcome() {

return <h1>Hello from Function Component</h1>;

}

With Hooks (e.g., useState), function components are now preferred for most development.

**6. Define Component Constructor**

In **Class Components**, the **constructor** is a special method used to initialize state and bind methods. It’s called once when the component is created.

**Syntax:**

jsx

CopyEdit

constructor(props) {

super(props);

this.state = {

name: 'React'

};

}

Key Notes:

* You must call super(props) before accessing this
* Useful for setting up initial state and binding event handlers

**7. Define render() Function**

The render() function is required in class components. It determines **what UI will be rendered** to the DOM.

**Example:**

jsx

CopyEdit

render() {

return <div>Welcome to React</div>;

}

* Must return JSX (React elements)
* Automatically called when state or props change
* Should be pure – avoid side effects inside render()

**OUTPUT:**

