# 3. ReactJS-HOL

## Concepts

### 1. Explain React Components

React components are the building blocks of any React application. They allow you to split the UI into independent, reusable pieces, and think about each piece in isolation. Components can be class-based or function-based and must return JSX that represents the UI.

### 2. Differences Between React Components and JavaScript Functions

- React components return JSX, while JavaScript functions return values or perform actions.  
- Components have lifecycle methods (in class components) and can manage state.  
- Components are used to build UI; JS functions are general-purpose logic blocks.

### 3. Types of Components

Class Components: ES6 classes that extend `React.Component`.  
Function Components: JavaScript functions that return JSX.  
Pure Components: Like class components but with shallow prop and state comparison.  
Higher-Order Components (HOC): Functions that take a component and return a new component.

### 4. Class Component

A class component is a JavaScript class that extends React.Component. It must include a render() method that returns JSX.

### 5. Function Component

A function component is a simpler way to write components using JavaScript functions. They return JSX directly and can use hooks like useState and useEffect to manage state and side-effects.

### 6. Component Constructor

The constructor is a method used in class components for initializing state and binding event handlers. It is called before the component is mounted.

### 7. render() Function

The render() function is required in class components. It describes what the UI should look like by returning JSX. It is automatically called whenever a component’s state or props change.

# Hands-On Lab:

# Student Management Portal - Score Calculator App

## Steps

## Step 1: Create React App

Run the following command in terminal:

npx create-react-app scorecalculatorapp

cd scorecalculatorapp

## Step 2: Create Components Folder and File

Under src, create a folder named Components and add a file CalculateScore.jsx with this code:

## CalculateScore.jsx Code

import React from "react";  
import "../Stylesheets/mystyle.css";  
  
function CalculateScore(props) {  
 const average = props.total / props.goal;  
  
 return (  
 <div className="score-container">  
 <h2>Student Score Calculator</h2>  
 <p><strong>Name:</strong> {props.name}</p>  
 <p><strong>School:</strong> {props.school}</p>  
 <p><strong>Total Marks:</strong> {props.total}</p>  
 <p><strong>Goals:</strong> {props.goal}</p>  
 <p><strong>Average Score:</strong> {average}</p>  
 </div>  
 );  
}  
  
export default CalculateScore;

## Step 3: Add Styles

Create Stylesheets folder under src and add mystyle.css with:

.score-container {  
 background-color: #f0f8ff;  
 padding: 20px;  
 border-radius: 8px;  
 width: 400px;  
 margin: auto;  
 margin-top: 50px;  
 font-family: Arial, sans-serif;  
 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
.score-container h2 {  
 color: #333;  
 margin-bottom: 15px;  
}

## Step 4: Edit App.jsx

import React from "react";  
import "./App.css";  
import CalculateScore from "./Components/CalculateScore";  
  
function App() {  
 return (  
 <div className="App">  
 <CalculateScore name="John Doe" school="XYZ High School" total={450} goal={5} />  
 </div>  
 );  
}  
  
export default App;

## Step 5: Run the Application

npm start

## Step 6: View in Browser

Open browser and go to: <http://localhost:3000>

**OUTPUT:**

