**OBJECTIVES**

1. **React Components**

* **What is a React Component?**
  + A component in React is a reusable, self-contained unit of code that represents part of a user interface (UI).
  + Think of components as building blocks of a React application.
* **Types of React Components**
  + **Functional Components**
    - Simple JavaScript functions.
    - Use React Hooks for state and lifecycle.
    - Example:

function Welcome() {

return <h1>Hello</h1>;

}

* + **Class Components**
    - Use ES6 class syntax and extend React.Component.
    - Can hold state and lifecycle methods (older way).
    - Example:

class Welcome extends React.Component {

render() {

return <h1>Hello</h1>;

}

}

* **Component Structure**
  + Each component:
    - Can accept **props** (inputs).
    - Can have its own **state** (data).
    - Returns JSX (UI structure).
* **Props (Properties)**
  + Used to pass data from parent to child components.
  + Read-only inside the child component.
  + Example:

function Greet(props) {

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

}

* **State**
  + Represents data that can change over time inside a component.
  + Managed using useState in functional components or this.state in class components.
  + Example:

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

* **Reusable**
  + Components can be reused throughout the app with different props.
* **Component Nesting**
  + Components can contain other components to build complex UIs.

1. **Identify the Differences Between Components and JavaScript Functions**

| **Aspect** | **JavaScript Function** | **React Component** |
| --- | --- | --- |
| Purpose | Executes logic or returns a value | Returns UI elements (JSX) |
| JSX Support | No | Yes (if it’s a component) |
| React Features | Cannot use Hooks or lifecycle methods | Can use state, props, hooks, and lifecycle methods |
| Reusability in UI | Not directly usable in React UI | Reusable in React JSX like <ComponentName /> |

1. **Identify the Types of Components**

React provides two main types of components:

1. **Class Components**
2. **Functional Components**
3. **Explain Class Component**

* A class component is a traditional way to define components using ES6 classes.
* Must extend React.Component and include a render() method.
* Can use state, lifecycle methods, and props.

1. **Explain Function Component**

* A function component is a simpler way to define components using JavaScript functions.
* Modern React encourages using function components with Hooks (e.g., useState, useEffect).

1. **Define Component Constructor**

* The constructor is a special method used in class components to:
  + Initialize state
  + Bind event handlers
  + Access props

1. **Define render() Function**

* The render() function is a mandatory method in class components.
* It returns JSX that describes what to display in the UI.