**11.ReactJS-HOL**

**1. Explain React events**  
React events are handled similarly to DOM events but follow the React-specific conventions. They are named using camelCase and passed as functions. React wraps the browser's native events in a cross-browser wrapper known as a SyntheticEvent to provide consistent behavior.

<button onClick={handleClick}>Click Me</button>

**2. Explain about event handlers**  
Event handlers in React are functions that get triggered in response to user interactions like clicks, typing, submitting a form, etc. These handlers are passed as props to the JSX elements.

function handleClick() {

alert('Button clicked!');

}

return <button onClick={handleClick}>Click Me</button>;

Handlers can be written inline or defined as separate functions.

**3. Define Synthetic event**  
SyntheticEvent is a cross-browser wrapper around the browser’s native event. It normalizes events so that they have consistent properties across different browsers. SyntheticEvent wraps all the common events like onClick, onChange, onSubmit, etc.

function handleChange(event) {

console.log(event.target.value); // event is a SyntheticEvent

}

<input type="text" onChange={handleChange} />

**4. Identify React event naming convention**

React uses camelCase for event names instead of lowercase (e.g., onClick instead of onclick).

Event handlers are passed as function references, not strings.

**Example:**

// Correct

<button onClick={handleClick}>Click</button>

// Incorrect

<button onclick="handleClick()">Click</button> // This will not work in React

This naming convention keeps React code consistent with JavaScript best practices.