## Objective Answers

### 1. Explain React Events

React events are the way to handle user interactions such as mouse clicks, keyboard input, form submissions, and other browser events within a React component. React wraps the native browser events into its own **SyntheticEvent** system to ensure cross-browser compatibility and improved performance. These events work similarly to JavaScript DOM events but follow a more consistent and declarative pattern. We can attach event handlers like onClick, onChange, or onSubmit directly within JSX, enabling interactive behavior in React apps.

### 2. Explain About Event Handlers

Event handlers in React are functions that are executed in response to user-triggered events. These functions are typically passed as props to JSX elements using attributes like onClick, onMouseOver, or onSubmit. For example, a button can execute a function when clicked:

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

In class components, we often bind event handlers using this in the constructor, whereas in functional components, we use arrow functions or hooks like useState and useEffect to manage event-driven state. Event handlers help React applications become dynamic and responsive to user actions.

### 3. Define Synthetic Event

A Synthetic Event is React’s cross-browser wrapper around the browser’s native event system. It wraps native DOM events and normalizes them so they behave consistently across all browsers. For example, a click event in React is a SyntheticEvent that behaves identically in Chrome, Firefox, and Internet Explorer. This abstraction helps React manage the event pool efficiently and reduces memory usage through event reuse. It also allows React to implement advanced features like batching updates and asynchronous rendering.

### 4. Identify React Event Naming Convention

React follows a camelCase naming convention for all event attributes instead of the lowercase used in standard HTML. For example:

HTML: <button onclick="myFunction()">Click</button>

React: <button onClick={myFunction}>Click</button>

All event handlers in React are passed as JavaScript functions (without parentheses unless we are explicitly calling them inside an arrow function). This consistent naming pattern helps differentiate React-specific syntax from raw HTML and maintains JavaScript code clarity within JSX.