Objective Explanation

Explain various ways of conditional rendering

React provides several ways to perform conditional rendering:

1. Using if/else statements: You can use standard JavaScript if/else to conditionally render components.

2. Ternary operator: A concise way to render based on a condition.

Example: {isLoggedIn ? <Logout /> : <Login />}

3. Logical && operator: Renders the component only if the condition is true.

Example: {isLoggedIn && <Dashboard />}

4. Switch-case statement: Useful for rendering one of many components based on state.

5. Immediately Invoked Function Expressions (IIFE): Wrap logic in a function and immediately execute.

Explain how to render multiple components

In React, you can render multiple components by:

1. Grouping in a parent component: Render multiple child components inside a single parent JSX tag.

Example:

```
<div>
  <Header />
  <MainContent />
  <Footer />
</div>
```

2. Using React Fragments: To avoid extra nodes in the DOM, use fragments.

Example:

```
<>
  <Component1 />
  <Component2 />
</>
```

Define list component

A list component in React is used to render a collection of similar elements from an array using JavaScript's `map()` method.

Example:

```
const items = ['Apple', 'Banana', 'Mango'];
const ItemList = () => (
 <ul>
  {items.map(item => <li key={item}>{item}</li>)}
```

```
  </ul>
);
```

Explain about keys in React applications

Keys are special string attributes used to identify which items in a list are changed, added, or removed.

- Helps React optimize rendering performance.
- Should be unique among siblings.
- Prefer using stable IDs as keys instead of array indexes.

Explain how to extract components with keys

When mapping over a list to create components, you can extract each item into its own component and pass the key as a prop.

Example:

```
function Item({ value }) {
  return <li>{value}</li>;
}
function ItemList({ items }) {
  return (
    <ul>
      {items.map(item => <Item key={item.id} value={item.name} />)}
    </ul>
  );
}
```

Explain React Map, map() function

The `map()` function in React is a JavaScript array method used to transform and render arrays of elements.

Usage:

```
items.map(item => <Component key={item.id} data={item} />)
```

- Commonly used in list rendering.
- Ensures each rendered component has a unique key.