Let's go through two examples of class components in React with detailed explanations:

Example 1: Counter Component

```jsx
import React, { Component } from 'react';

class Counter extends Component {
  constructor(props) {
    super(props);
    // Initialize state with count set to 0
    this.state = {
      count: 0
    };
  }

  // Event handler for incrementing count
  handleClick = () => {
    // Update count by 1 using setState
    this.setState(prevState => ({
      count: prevState.count + 1
    }));
  }

  render() {
    // Render the current count and a button to increment it
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={this.handleClick}>Increment</button>
      </div>
    );
  }
}

export default Counter;
```

Explanation:
- This class component is named `Counter` and extends `Component` from React.
- In the constructor, `this.state` is used to initialize the component's state with a property `count` set to `0`.
- `handleClick` is a class method that increments the count when called. It uses `this.setState()` to update the state with the new count value.

- In the `render` method, the current count is displayed along with a button. When the button is clicked, `handleClick` method is invoked to update the count.

Example 2: UserList Component

```jsx
import React, { Component } from 'react';

class UserList extends Component {
  constructor(props) {
    super(props);
    // Initialize state with an empty array for users
    this.state = {
      users: []
    };
  }

  // Simulating fetching users from an API
  componentDidMount() {
    fetch('https://jsonplaceholder.typicode.com/users')
      .then(response => response.json())
      .then(users => this.setState({ users }))
      .catch(error => console.error('Error fetching users:', error));
  }

  render() {
    // Render the list of users
    return (
      <div>
        <h2>User List</h2>
        <ul>
          {this.state.users.map(user => (
            <li key={user.id}>{user.name}</li>
          ))}
        </ul>
      </div>
    );
  }
}

export default UserList;
```

Explanation:

- This class component is named `UserList` and extends `Component` from React.
- In the constructor, `this.state` is used to initialize the component's state with an empty array `users`.
- The `componentDidMount` lifecycle method is used to fetch user data from an API (in this example, JSONPlaceholder) after the component mounts. Once the data is fetched, it updates the state with the retrieved users.
- In the `render` method, it maps over the `users` array in the state and renders a list of user names. Each user is displayed as a list item with a unique `key` attribute.

These examples demonstrate how class components in React manage state and lifecycle methods to build interactive and dynamic user interfaces.