# ReactJS Hands-On Lab 11 - Solutions

## 1. Creating the React App

Run the following command to create the React app:  
npx create-react-app eventexamplesapp  
cd eventexamplesapp  
code .

## 2. Event Handling and Multiple Methods

Inside src/App.js:  
import React from 'react';  
  
class App extends React.Component {  
 constructor(props) {  
 super(props);  
 this.state = {  
 count: 0,  
 message: ""  
 };  
 }  
  
 increment = () => {  
 this.setState({ count: this.state.count + 1 });  
 this.sayHello();  
 };  
  
 decrement = () => {  
 this.setState({ count: this.state.count - 1 });  
 };  
  
 sayHello = () => {  
 this.setState({ message: "Hello! Counter updated." });  
 };  
  
 sayWelcome = (msg) => {  
 this.setState({ message: msg });  
 };  
  
 handleClick = (event) => {  
 this.setState({ message: "I was clicked" });  
 };  
  
 render() {  
 return (  
 <div>  
 <h2>Event Handling Example</h2>  
 <p>Count: {this.state.count}</p>  
 <button onClick={this.increment}>Increment</button>  
 <button onClick={this.decrement}>Decrement</button>  
 <button onClick={() => this.sayWelcome("Welcome!")}>Say Welcome</button>  
 <button onClick={this.handleClick}>OnPress</button>  
 <p>{this.state.message}</p>  
 </div>  
 );  
 }  
}  
  
export default App;

## 3. CurrencyConvertor Component

Inside src/CurrencyConvertor.js:  
import React from 'react';  
  
class CurrencyConvertor extends React.Component {  
 constructor(props) {  
 super(props);  
 this.state = {  
 rupees: "",  
 euros: ""  
 };  
 }  
  
 handleChange = (e) => {  
 this.setState({ rupees: e.target.value });  
 };  
  
 handleSubmit = () => {  
 const euroRate = 0.011; // Example conversion rate  
 this.setState({ euros: (this.state.rupees \* euroRate).toFixed(2) });  
 };  
  
 render() {  
 return (  
 <div>  
 <h2>Currency Convertor</h2>  
 <input type="number" value={this.state.rupees} onChange={this.handleChange} placeholder="Enter Rupees" />  
 <button onClick={this.handleSubmit}>Convert</button>  
 <p>Euros: €{this.state.euros}</p>  
 </div>  
 );  
 }  
}  
  
export default CurrencyConvertor;

## 4. Using Both Components in App.js

You can import and render the CurrencyConvertor in App.js like this:  
import CurrencyConvertor from './CurrencyConvertor';  
  
...inside render() or return():  
<CurrencyConvertor />

## 5. Output Layout

The app demonstrates event handling including multiple method calls, parameter passing, synthetic event, and form handling for currency conversion.