

// Question 1: What are components in React? Explain the difference between functional components and class components.

/*

Components in React are independent, reusable pieces of UI that define how a part of the UI should appear.

React components can be functional or class-based

Functional Components:

- Simpler and typically used for presenting UI.
- They are stateless and are defined as functions.
- Example:

*/

```
const MyComponent = () => <h1>Hello, World!</h1>
```

/*

Class Components:

- More feature-rich, can hold local state and lifecycle methods.
- Defined as ES6 classes.
- Example:

*/

```
class MyComponent extends React.Component {  
  render() {  
    return <h1>Hello, World!</h1>;  
  }  
}
```

// Question 2: How do you pass data to a component using props?

/*

Props (short for properties) are used to pass data from a parent component to a child component in React.

Example:

```
*/
```

```
const Greeting = (props) => <h1>Hello, {props.name}!</h1>;
```

```
// Usage of the component
```

```
<Greeting name="John" />
```

```
// Question 3: What is the role of render() in class components?
```

```
/*
```

The render() method in class components returns the JSX that should be displayed on the screen.

It is required for any class component.

```
*/
```

```
class MyComponent extends React.Component {
```

```
  render() {
```

```
    return <h1>Hello, World!</h1>;
```

```
  }
```

```
}
```

```
// LAB EXERCISE
```

```
// Task 1: Create a functional component Greeting that accepts a name as a prop and displays "Hello, [name]!".
```

```
const Greeting = (props) => {
```

```
  return <h1>Hello, {props.name}!</h1>;
```

```
};
```

```
// Usage
```

```
<Greeting name="Alice" />
```

// Task 2: Create a class component WelcomeMessage that displays "Welcome to React!" and a render() method.

```
class WelcomeMessage extends React.Component {  
  render() {  
    return <h1>Welcome to React!</h1>;  
  }  
}
```

// Usage

```
<WelcomeMessage />
```

// Question 1: What are props in React.js? How are props different from state?

/*

Props are used to pass data from parent to child components. They are immutable.

State is used to store data within a component and can be changed within the component using setState() (for class components) or useState() (for functional components).

*/

// Question 2: Explain the concept of state in React and how it is used to manage component data.

/*

State in React is used to store data that can change over time. State allows a component to react to changes, such as user interactions, network responses, or timer events.

*/

```
class Counter extends React.Component {  
  constructor() {  
    super();  
    this.state = { count: 0 };  
  }  
}
```

```
}
```

```
increment = () => {  
  this.setState({ count: this.state.count + 1 });  
};
```

```
render() {  
  return (  
    <div>  
      <h1>{this.state.count}</h1>  
      <button onClick={this.increment}>Increment</button>  
    </div>  
  );  
}  
}
```

// Question 3: Why is this.setState() used in class components, and how does it work?

```
/*  
  this.setState() is used to update the state of a component. It triggers a re-render of the component  
  with the updated state.  
  
  It is asynchronous, meaning React batches updates to optimize performance.  
*/
```

// LAB EXERCISE

// Task 1: Create a React component UserCard that accepts name, age, and location as props and displays them in a card format.

```
const UserCard = (props) => {  
  return (  
    <div className="card">
```

```
    <h2>{props.name}</h2>
    <p>Age: {props.age}</p>
    <p>Location: {props.location}</p>
  </div>
);
};

// Usage
<UserCard name="John" age={30} location="New York" />
```

// Task 2: Create a Counter component with a button that increments a count value using React state. Display the current count on the screen.

```
class Counter extends React.Component {
  constructor() {
    super();
    this.state = { count: 0 };
  }

  increment = () => {
    this.setState({ count: this.state.count + 1 });
  };

  render() {
    return (
      <div>
        <h1>{this.state.count}</h1>
        <button onClick={this.increment}>Increment</button>
      </div>
    );
  }
}
```

