✓ Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

1. What of the below scenarios are valid for choosing context instead of local state? Select all that apply.

1 / 1 point

☑ The locale or language that should be used in the application's text.

✓ **Correct**
That's correct, since that's global state that affects all component's text.

□ The current selection of a group of radio buttons.

☑ The visibility state of an alert that overlays into the whole application.

✓ **Correct**
That's correct, global alerts are well suited for context since any component can trigger them.

2. What is the problem of props drilling? Select all that apply.

1 / 1 point

☑ Components having to pass down props all the way to the children that need to consume them.

✓ **Correct**
That's correct, components drill down props across several levels to reach the children that need them.

□ Components not knowing the local state of their parents.

☑ Components receiving more props than they should.

✓ **Correct**
That's correct, props drilling involves components receiving props that they don't use or need at all.

3. When creating a new piece of application state, what is the bare minimum of React APIs you would need to define it?

1 / 1 point

● Context and local state.

○ Context and props.

○ Context, props and local state.

✓ **Correct**
That's correct, you need local state to define the global state and Context to inject it into a tree.

4. What happens when the `value` prop of the Context Provider changes?

1 / 1 point

● All the consumer components re-render with the updated value.

○ The Context Provider component gets recreated.

○ The whole component tree under the Context Provider gets re-rendered.

✓ **Correct**
That's correct, all components subscribed to that context get updated.

5. What happens when you wrap a component with the `React.memo` API, such as `React.memo(Component)`. Select all that apply.

1 / 1 point

☑ Whether the component should re-render could be determined by some custom logic that uses the previous props and the current props.

Correct
That's correct, you can provide an additional function that receives previous props and current props to determine if an update should occur.

☐ The component never gets updated no matter if there was a change in its local state or the props it receives.

☑ React provides a performance optimization.

Correct
That's correct, it avoids unnecessary re-renders automatically.