Congratulations! You passed!

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

Go to next item

1. When dealing with cross-cutting data in your React applications, what are some of the problems of using a custom hook to encapsulate that logic? Select all that apply. **1 / 1 point**

- [ ] There are no problems at all with hooks, being the best suited tool for the job.
- [x] The fact that you would have to alter the implementation of each component that needs that specific data.

Correct
That's correct, you would have to add the custom hook to each particular component that needs that data.

- [x] That it turns a stateless component into a stateful one.

Correct
That's correct, the addition of a custom hook that deals with specific data introduces state into the component.

2. Here, you can find the APIs of some higher-order components that have been already implemented. Amongst all the options, which ones present an invalid signature that doesn't follow the convention? Select all that apply. **1 / 1 point**

- ( )
```
withSubscription(Component, options)
```
- (•)
```
withSubscription(() => getData(), Component)
```
- ( )
```
withSubscription(() => getData())(Component)
```

Correct
That's correct, the component should come as the first argument and any additional parameters should go afterwards.

3. What are some of the best practices to follow when implementing the higher-order component pattern? Select all that apply. **1 / 1 point**

- [x] Maximize composability.

Correct
That's correct, using proper API definitions allow you to combine different HOCs in a composable manner.

- [x] Passed unrelated props through to the Wrapped Component.

Correct
That's correct, you should pass all the other props through to the wrapped component.

- [ ] Always use HOCs and create your enhanced components inside other components.
- [ ] Mutate the original component

4. What are some of the differences between higher-order components and `render` props? Select all that apply. **1 / 1 point**

- [x] Render props provide the new data as a function parameter, whereas components wrapped with an HOC get the new data as a new prop.

  Correct
  That's correct, `render` props do so dynamically inside the JSX and HOCs provide an extra prop.

- [x] They inject the new props in the component to be enhanced in a different way.

  Correct
  That's correct, `render` props dynamically inside the JSX and HOCs as an extra prop.

- [ ] Higher-order components modify the original implementation of the component, whereas the Render Props pattern doesn't.

5. True or false. A component with a `render` prop as renderer can do anything a higher-order component can do. **1 / 1 point**

- (•) True
- ( ) False

Correct
That's correct, they are just two different implementations for encapsulating cross-cutting concerns, but they serve the same purpose.