1.

Question 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.

There are no problems at all with hooks, being the best suited tool for the job.

That it turns a stateless component into a stateful one.

The fact that you would have to alter the implementation of each component that needs that specific data.

2.

Question 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.

withSubscription(() => getData())(Component)

withSubscription(Component, options)

withSubscription(() => getData(), Component)

3.

Question 3

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

Mutate the original component

Passed unrelated props through to the Wrapped Component.

Maximize composability.

Always use HOCs and create your enhanced components inside other components.

4.

Question 4

What are some of the differences between higher-order components and render props? Select all that apply.

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

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

They inject the new props in the component to be enhanced in a different way.

5.

Question 5

True or false. A component with a render prop as renderer can do anything a higher-order component can do.

False

True