1.

Question 1

How is array destructuring relevant to hooks in React?

It is a way to get individual items from an array of items, and save those individual items as separate components.

It makes it possible to handle click events.

It makes it possible to reassign state objects.

It makes the Virtual DOM possible.

2.

Question 2

Is the following paragraph correct?

With array destructuring, you are free to give any variable name to the items that you destructure from an array. Contrary to that, when destructuring objects, you have to destructure a property of an object using that exact property's name as the name of the destructured variable.

Yes

No

3.

Question 3

The useEffect hook is a way to:

handle one-way data flows

handle visual effects (animations and transitions) in React

handle a side effect.

4.

Question 4

Which answer is correct about the following code snippet?

useEffect( () => {

if (data !== '') {

setData('test data')

}

})

This code is ok, except the fact that you should replace the if statement with a ternary operator.

This code is breaking the rules of hooks

This code is not breaking the rules of hooks

5.

Question 5

Choose an example of a side-effect with which you’d need to use a useEffect hook:

Run a Fetch API call in React.

Render some prop values on the screen.

Update the value of the state variable in a child component.

6.

Question 6

Complete the sentence:

The useState hook starts with an initial state, but...

the useReducer hook gets a reducer function in addition to the initial state.

the useReducer hook cannot be used to track the state at all.

the userReducer hook must be used when the initial state is an object.

7.

Question 7

True or false: useRef is a custom hook in React.

Yes.

No

8.

Question 8

JavaScript is a single-threaded language, meaning...

...it can only do a single thing at any given time.

...you can use it with React only when this single thread is used with the useEffect hook.

...you can use it with React only when this single thread is passed to the useState variable.

9.

Question 9

Which statement is correct about the following code snippet:

import { useEffect } from "react";

function useConsoleLog(varName) {

useEffect(() => {

console.log(varName);

});

}

export default useConsoleLog;

Choose the correct statement below.

This is an example of a custom hook.

This code is an example of an implicit state-updating function.

This code is an example of an explicit state-updating function.

10.

Question 10

Find the error in this code:

import {useState} from "react";

export default function App() {

const [restaurantName, setRestaurantName] = useState("Lemon");

function updateRestaurantName() {

useRestaurantName("Little Lemon");

};

return (

<div>

<h1>{restaurantName}</h1>

<button onClick={updateRestaurantName}>

Update restaurant name

</button>

</div>

);

};

The onClick event should not be this: onClick={updateRestaurantName}

The state-setting function's variable name in the array destructuring should not be setRestaurantName.

The code inside the updateRestaurantName() function definition should not invoke useRestaurantName("Little Lemon")