Q1. If you want to import just the Component from the React library, what syntax do you use?

- import React.Component from ‘react’
- import [Component] from ‘react’
- import Component from ‘react’
- **import { Component } from ‘react’**

Q2. If a function component should always render the same way given the same props, what is a simple performance optimization available for it?

- **Wrap it in the React.memo higher-order component.**
- Implement the useReducer Hook.
- Implement the useMemo Hook.
- Implement the shouldComponentUpdate lifecycle method.

Q3. How do you fix the syntax error that results from running this code?

```
const person =(firstName, lastName) =>
{
first: firstName,
last: lastName
}
console.log(person(“Jill”, “Wilson”))
```

- **Wrap the object in parentheses.**
- Call the function from another file.
- Add a return statement before the first curly brace.
- Replace the object with an array.

Q4. If you see the following import in a file, what is being used for state management in the component?

```
import React, {useState} from ‘react’;
```

- **React Hooks**
- stateful components
- math
- class components

Q5. Using object literal enhancement, you can put values back into an object. When you log person to the console, what is the output?

```
const name = ‘Rachel’;
```

const age = 31;
const person = { name, age };
console.log(person);

- {{name: “Rachel”, age: 31}}
- **{name: “Rachel”, age: 31}**
- {person: “Rachel”, person: 31}}
- {person: {name: “Rachel”, age: 31}}

Q6. What is the testing library most often associated with React?

- Mocha
- Chai
- Sinon
- **Jest**

Q7. To get the first item from the array (“cooking”) using array destructuring, how do you adjust this line?

- const topics = [‘cooking’, ‘art’, ‘history’];
- const first = [“cooking”, “art”, “history”]
- const [] = [“cooking”, “art”, “history”]
- const [, first][“cooking”, “art”, “history”]
- **const [first] = [“cooking”, “art”, “history”]**

Q8. How do you handle passing through the component tree without having to pass props down manually at every level?

- React Send
- React Pinpoint
- React Router
- **React Context**

Q9. What should the console read when the following code is run?
const [, , animal] = [‘Horse’, ‘Mouse’, ‘Cat’];
console.log(animal);

- Horse
- **Cat**
- Mouse
- undefined

10. What is the name of the tool used to take JSX and turn it into createElement calls?

- JSX Editor
- ReactDOM
- Browser Buddy
- **Babel**

11. Why might you use useReducer over useState in a React component?

- when you want to replace Redux
- **when you need to manage more complex state in an app**
- when you want to improve performance
- when you want to break your production app

12. Which props from the props object is available to the component with the following syntax?
<Message {...props} />

- any that have not changed
- **all of them**
- child props
- any that have changed

13. Consider the following code from React Router. What do you call :id in the path prop?
<Route path="/:id" />

- This is a route modal
- **This is a route parameter**
- This is a route splitter
- This is a route link

14. If you created a component called Dish and rendered it to the DOM, what type of element would be rendered?

```
function Dish() {
return <h1>Mac and Cheese</h1>;
}
```

- ReactDOM.render(<Dish />, document.getElementById('root'));
- div

- section
- component
- **h1**

15. What does this React element look like given the following function? (Alternative: Given the following code, what does this React element look like?)

React.createElement('h1', null, "What's happening?");

- <h1 props={null}>What's happening?</h1>
- **<h1>What's happening?</h1>**
- <h1 id="component">What's happening?</h1>
- <h1 id="element">What's happening?</h1>

16. What property do you need to add to the Suspense component in order to display a spinner or loading state?

```
function MyComponent() {
return (
<Suspense>
<div>
<Message />
</div>
</Suspense>
);
}
```

- lazy
- loading
- **fallback**
- spinner

17. What do you call the message wrapped in curly braces below?

```
const message = 'Hi there';
const element = <p>{message}</p>;
```

- a JS function
- a JS element
- **a JS expression**
- a JSX wrapper

18. What can you use to handle code splitting?

- React.memo
- React.split
- **React.lazy**
- React.fallback

19. When do you use useLayoutEffect?

- to optimize for all devices
- to complete the update
- to change the layout of the screen
- **when you need the browser to paint before the effect runs**

20. What is the difference between the click behaviors of these two buttons (assuming that this.handleClick is bound correctly)?

A. <button onClick={this.handleClick}>Click Me</button>

B. <button onClick={event => this.handleClick(event)}>Click Me</button>

- Button A will not have access to the event object on click of the button.
- Button B will not fire the handler this.handleClick successfully.
- Button A will not fire the handler this.handleClick successfully.
- **There is no difference.**

21. How do you destructure the properties that are sent to the Dish component?

```
function Dish(props) {
return (
<h1>
{props.name} {props.cookingTime}
</h1>
);
}
```

- function Dish([name, cookingTime]) { return <h1>{name} {cookingTime}</h1>; }
- **function Dish({name, cookingTime}) { return <h1>{name} {cookingTime}</h1>; }**
- function Dish(props) { return <h1>{name} {cookingTime}</h1>; }
- function Dish(...props) { return <h1>{name} {cookingTime}</h1>; }

22. When might you use React.PureComponent?

- when you do not want your component to have props
- when you have sibling components that need to be compared
- **when you want a default implementation of shouldComponentUpdate()**
- when you do not want your component to have state

23. Why is it important to avoid copying the values of props into a component's state where possible?

- because you should never mutate state
- because getDerivedStateFromProps() is an unsafe method to use
- **because you want to allow a component to update in response to changes in the props**
- because you want to allow data to flow back up to the parent

24. What is the children prop?

- a property that adds child components to state
- **a property that lets you pass components as data to other components**
- a property that lets you set an array as a property
- a property that lets you pass data to child elements

25. Which attribute do you use to replace innerHTML in the browser DOM?

- injectHTML
- **dangerouslySetInnerHTML**
- weirdSetInnerHTML
- strangeHTML

26. Which of these terms commonly describe React applications?

- **declarative**
- integrated
- closed
- imperative

27. When using webpack, why would you need to use a loader?

- to put together physical file folders
- **to preprocess files**
- to load external data

- to load the website into everyone’s phone

28. A representation of a user interface that is kept in memory and is synced with the “real” DOM is called what?

- **virtual DOM**
- DOM
- virtual elements
- shadow DOM

29. You have written the following code but nothing is rendering. How do you fix this problem?

```
const Heading = () => {
<h1>Hello!</h1>;
};
```

- Add a render function.
- **Change the curly braces to parentheses or add a return statement before the h1 tag.**
- Move the h1 to another component.
- Surround the h1 in a div.

Q30. To create a constant in JavaScript, which keyword do you use?

- **const**
- let
- constant
- var

Q31. What do you call a React component that catches JavaScript errors anywhere in the child component tree?

- error bosses
- error catchers
- error helpers
- **error boundaries**

Q32. In which lifecycle method do you make requests for data in a class component?

- constructor
- **componentDidMount**

- componentWillReceiveProps
- componentWillMount

Q33. React components are composed to create a user interface. How are components composed?

- by putting them in the same file
- **by nesting components**
- with webpack
- with code splitting

Q34. All React components must act like **__** with respect to their props.

- monads
- **pure functions**
- recursive functions
- higher-order functions

Q35. Why might you use a ref?

- **to directly access the DOM node**
- to refer to another JS file
- to call a function
- to bind the function

Q36. What is [e.target.id] called in the following code snippet?

```
handleChange(e) {
this.setState({ [e.target.id]: e.target.value })
}
```

- a computed property name
- a set value
- **a dynamic key**
- a JSX code string

Q37. What is the name of this component?

```
class Clock extends React.Component {
render() {
return <h1>Look at the time: {time}</h1>;
}
}
```

- **Clock**
- It does not have a name prop.
- React.Component
- Component

Q38. What is sent to an Array.map() function?

- **a callback function that is called once for each element in the array**
- the name of another array to iterate over
- the number of times you want to call the function
- a string describing what the function should do

Q39. Why is it a good idea to pass a function to setState instead of an object?

- It provides better encapsulation.
- It makes sure that the object is not mutated.
- It automatically updates a component.
- **setState is asynchronous and might result in out of sync values.**

Q40. What package contains the render() function that renders a React element tree to the DOM?

- **React**
- ReactDOM
- Render
- DOM

Q41. How do you set a default value for an uncontrolled form field?

- Use the value property.
- **Use the defaultValue property.**
- Use the default property.
- It assigns one automatically.

Q42. What do you need to change about this code to get it to run?
class clock extends React.Component {
render() {
return <h1>Look at the time: {this.props.time}</h1>;
}
}

- Add quotes around the return value
- Remove this
- Remove the render method
- **Capitalize clock**

Q43. Which Hook could be used to update the document's title?

- **useEffect(function updateTitle() { document.title = name + ' ' + lastname; });**
- useEffect(() => { title = name + ' ' + lastname; });
- useEffect(function updateTitle() { name + ' ' + lastname; });
- useEffect(function updateTitle() { title = name + ' ' + lastname; });

Q44. What can you use to wrap Component imports in order to load them lazily?

- React.fallback
- React.split
- **React.lazy**
- React.memo

Q45. How do you invoke setDone only when component mounts, using hooks?

```
function MyComponent(props) {
const [done, setDone] = useState(false);
return <h1>Done: {done}</h1>;
}
```

- useEffect(() => { setDone(true); });
- **useEffect(() => { setDone(true); }, []);**
- useEffect(() => { setDone(true); }, [setDone]);
- useEffect(() => { setDone(true); }, [done, setDone]);

Q46. What value of button will allow you to pass the name of the person to be hugged?

```
class Huggable extends React.Component {
hug(id) {
console.log("hugging " + id);
}
render() {
let name = "kitteh";
```

let button = // Missing Code
return button;
}
}

- **<button onClick={(name) => this.hug(name)>Hug Button</button>**
- <button onClick={this.hug(e, name)}>Hug Button</button>
- <button onClick={(e) => hug(e, name)}>Hug Button</button>
- <button onClick={(e) => this.hug(name, e)}>Hug Button</button>

Q47. Currently, handleClick is being called instead of passed as a reference. How do you fix this?
<button onClick={this.handleClick()}>Click this</button>

- <button onClick={this.handleClick.bind(handleClick)}>Click this</button>
- <button onClick={handleClick()}>Click this</button>
- **<button onClick={this.handleClick}>Click this</button>**
- <button onclick={this.handleClick}>Click this</button>

Q48. Which answer best describes a function component?

- A function component is the same as a class component.
- **A function component accepts a single props object and returns a React element.**
- A function component is the only way to create a component.
- A function component is required to create a React component.

Q49. Which library does the fetch() function come from?

- FetchJS
- ReactDOM
- **No library. fetch() is supported by most browsers.**
- React

Q50. What will happen when this useEffect Hook is executed, assuming name is not already equal to John?
useEffect(() => {
setName('John');
}, [name]);

- It will cause an error immediately.

- It will execute the code inside the function, but only after waiting to ensure that no other component is accessing the name variable.
- **It will update the value of name once and not run again until name is changed from the outside.**
- It will cause an infinite loop.

Q51. Which choice will not cause a React component to rerender?

- if the component calls this.setState(...)
- the value of one of the component's props changes
- if the component calls this.forceUpdate()
- **one of the component's siblings rerenders**

Q52. You have created a new method in a class component called handleClick, but it is not working. Which code is missing?

```
class Button extends React.Component{
constructor(props) {
super(props);
// Missing line
}handleClick() {...}
}
```

- this.handleClick.bind(this);
- props.bind(handleClick);
- this.handleClick.bind();
- **this.handleClick = this.handleClick.bind(this);**

Q53. React does not render two sibling elements unless they are wrapped in a fragment. Below is one way to render a fragment. What is the shorthand for this?

- C

Copy

```
<>

  <h1>Our Staff</h1>

  <p>Our staff is available 9-5 to answer your questions</p>
```

</>

Q54. If you wanted to display the count state value in the component, what do you need to add to the curly braces in the h1?

```
class Ticker extends React.component {
constructor(props) {
super(props);
this.state = { count: 0 };
}
render() {
return <h1>{}</h1>;
}
}
```

- **this.state.count**
- count
- state
- state.count

Q55. Per the following code, when is the Hello component displayed?

```
const greeting = isLoggedIn ? <Hello /> : null;
```

- never
- **when is LoggedIn is true**
- when a user logs in
- when the Hello function is called

Q56. In the following code block, what type is orderNumber?

```
ReactDOM.render(<Message orderNumber="16" />,
document.getElementById('root'));
```

- **string**
- boolean
- object
- number

Q57. You have added a style property to the h1 but there is an unexpected token error when it runs. How do you fix this?

```
const element = <h1 style={ backgroundColor: "blue" }>Hi</h1>;
```

- const element = <h1 style=”backgroundColor: “blue””}>Hi</h1>;
- **const element = <h1 style={{backgroundColor: “blue”}}>Hi</h1>;**
- const element = <h1 style={blue}>Hi</h1>;
- const element = <h1 style=”blue”>Hi</h1>;

Q58. Which function is used to update state variables in a React class component?

- replaceState
- refreshState
- updateState
- **setState**

Q59. Consider the following component. What is the default color for the star?
const Star = ({ selected = false }) => <Icon color={selected ? ‘red’ : ‘grey’} />;

- black
- red
- **grey**
- white

Q60. Which answer best describes a function component?(Not sure answer)

- A function component is the same as a class component.
- **A function component accepts a single props object and returns a React element.**
- A function component is the only way to create a component.
- A function component is required to create a React component.

Q61.Which library does the fetch() function come from?

- FetchJS
- ReactDOM
- **No library. fetch() is supported by most browsers.**
- React

Q62.What is the difference between the click behaviors of these two buttons(assuming that this.handleClick is bound correctly)
A. <button onClick=this.handleClick>Click Me</button>
B. <button onClick={event => this.handleClick(event)}>Click Me</button>

- Button A will not have access to the event object on click of the button
- Button A will not fire the handler this.handleClick successfully
- There is no difference
- Button B will not fire the handler this.handleClick successfully

Q63.What will happen when this useEffect Hook is executed, assuming name is not already equal to John?
useEffect(() => {
setName(‘John’);
}, [name]);

- It will cause an error immediately.
- It will execute the code inside the function, but only after waiting to ensure that no other component is accessing the name variable.
- **It will update the value of name once and not run again until name is changed from the outside.**
- It will cause an infinite loop.

Q64. How would you add to this code, from React Router, to display a component called About?
<Route path=”/:id” />

- javascript (<Route path=”/:id”> {‘ ’} <About /> </Route>)”` ”” ””`; ”””; ”””`
- javascript (<Route path=”/tid” about={Component} />)”` ”” ””`; ”””; ”””`
- javascript (<Route path=”/:id” route={About} />)”` ”” ””`; ”””; ”””`
- javascript (<Route> <About path=”/:id” /> </Route>)”` ”” ””`; ”””; ”””`

Q65. Which class-based component is equivalent to this function component?
const Greeting ({ name }) > <h1>Hello {name}!</h1>;

- javascript class Greeting extends React.Component { constructor() { return <h1>Hello (this.props.name)!</h1>; } } ”` ”” ””`; ”””; ”””`
- javascript class Greeting extends React.Component { <h1>Hello (this.props.name}!</h1>; } ”` ”” ””` ””” ”””`
- javascript class Greeting extends React.Component { return <h1>Hello (this.props.name) 1</h1>; } ”` ”” ””` ””” ”””`
- javascript class Greeting extends React.Component (render({ name }) { return <h1>Hello (name)} !</h1>; }) ”` ”” ””` ””” ”””`

Q66. Give the code below, what does the second argument that is sent to the render function describe?
ReactDOM.render(
<h1>Hi<h1>,
document.getElementById(‘root’)
)

- **where the React element should be added to the DOM**
- where to call the function
- where the root component is
- where to create a new JavaScript file

Q67. Why should you use React Router’s Link component instead of a basic <a> tag in React?

- The link component allows the user to use the browser’s Back button.
- There is no difference–the Link component is just another name for the <a> tag.
- The <a> tag will cause an error when used in React.
- The <a> tag triggers a full page reload, while the Link component does not.

Q68. What is the first argument, x, that is sent to the createElement function?
React.createElement(x, y, z);

- **the element that should be created**
- the order in which this element should be placed on the page
- the properties of the element
- data that should be displayed in the element

Q69. Which class-based lifecycle method would be called at the same time as this effect Hook?
useEffect(() => {
// do things
}, []);

- componentWillUnmount
- **componentDidMount**
- render

- componentDidUpdate

Q70. Given the code below, what does the second argument that is sent to the render function describe?

```
ReactDOM.render(
<h1>Hi</h1>
document.getElementById(‘root’)
);
```

- **where the React element should be added to the DOM**
- where to call the function
- where the root component is
- where to create a new JavaScript file

Q71. What is the first argument, x, that is sent to the createElement function?

```
React.createElement(x,y,z);
```

- **the element that should be created**
- the order in which this element should be placed on the page
- the properties of the element
- data that should be displayed in the element.