React Fundamentals II Quiz

ChatGPT: https://chatgpt.com/c/68240169-db1c-8009-bc5c-9ffc0a851812

Here's a comprehensive 40-question exam based on the article "Learn React – A Guide to the Key Concepts" from freeCodeCamp:

https://www.freecodecamp.org/news/learn-react-key-concepts/

- React Key Concepts Exam
- SECTION A: Multiple Choice (10 Questions)
 - 1. What is the purpose of React?
 - o A) Server-side rendering
 - o B) Managing backend databases
 - o C) Building user interfaces
 - o D) Writing stylesheets
 - 2. Which of the following best describes a React component?
 - o A) A CSS class
 - o B) A JavaScript function that returns JSX
 - o C) A backend API
 - o D) An HTML template
 - 3. Which syntax is used to define JSX?
 - o A) HTML-like
 - o B) CSS
 - o C) XML-like JavaScript
 - o D) JSON
 - 4. What is required when returning multiple elements from a component?
 - o A) Using a class
 - o B) Wrapping them in a parent element or fragment
 - o C) Adding a semicolon

	0	D) Using setTimeout
5.	What	hook allows you to store local state in a function component?
	0	A) useReducer
	0	B) useEffect
	0	C) useState
	0	D) useMemo
6.	React	's virtual DOM improves performance by:
	0	A) Writing faster CSS
	0	B) Directly manipulating the DOM
	0	C) Minimizing DOM updates
	0	D) Rendering the entire page every time
7.	What	happens when you call setState?
	0	A) A page refresh
	0	B) DOM is cleared
	0	C) The component is re-rendered
	0	D) Component is deleted
8.	Keys i	n React lists are important because:
	0	A) They act as passwords
	0	B) They help React identify elements that changed
	0	C) They are required by browsers

o D) They make JSX readable

o A) useRef

o B) useMemo

o C) useEffect

o D) useCallback

9. Which hook is best for performing side effects?

- 10. What is the difference between props and state?
- A) props are mutable; state is not
- B) state is passed to children; props are internal
- C) props are passed from parent; state is internal
- D) There is no difference

✓ SECTION B: True/False (5 Questions)

- 11. JSX is valid JavaScript.
- 12. You must use class components to manage state in React.
- 13. React components must begin with a capital letter.
- 14. The useEffect hook runs only once by default.
- 15. You can return multiple elements in a React component without a parent wrapper.

SECTION C: Short Answer (10 Questions)

- 16. What is JSX and why is it useful in React?
- 17. Describe the concept of "lifting state up" in React.
- 18. Why should you avoid modifying state directly?
- 19. When would you use use Effect with an empty dependency array?
- 20. What are fragments and why might you use them?
- 21. How can props be used to make components reusable?
- 22. Why is immutability important in React state?
- 23. What does the key prop do in a .map() list rendering?
- 24. Explain the concept of "one-way data flow" in React.
- 25. When using useState, what does the setter function actually do?

SECTION D: Code Analysis (5 Questions)

```
26. What does the following code render?
```

```
jsx
CopyEdit
function Hello() {
return <h1>Hello World</h1>;
}
   27. What will be the output and explanation?
jsx
CopyEdit
function Counter() {
const [count, setCount] = useState(0);
return (
 <div>
  {count}
  <button onClick={() => setCount(count + 1)}>Increment/button>
 </div>
);
}
   28. Find and correct the error:
jsx
CopyEdit
function App() {
const [name, setName] = useState;
return {name};
}
   29. What will useEffect(() => console.log('Hi')) do in this code?
```

```
jsx
CopyEdit
function Greeting() {
  useEffect(() => console.log('Hi'));
  return Welcome;
}
  30. What is wrong with this code and how can you fix it?
jsx
CopyEdit
function List() {
  const items = ['A', 'B', 'C'];
  return items.map((item) => {i>{item}};
}
```

SECTION E: Critical Thinking / Extended Response (5 Questions)

- 31. Compare and contrast class components vs function components in React.
- 32. Describe a scenario where you would lift state up and explain why it's necessary.
- 33. Why is useEffect sometimes referred to as the replacement for lifecycle methods like componentDidMount?
- 34. Discuss a situation where using **memoization** (useMemo, useCallback) is beneficial.
- 35. In a component where you fetch data from an API, describe how you'd prevent memory leaks or race conditions using useEffect.

36. Draw a React component tree for the following layout:		
	A Dashboard contains a Sidebar and MainContent. MainContent has two children:	
	Header and Content.	

37. **Draw a flow diagram** showing how props flow from a parent to multiple children and how a child might lift a value up via an event handler.

***	SECTION G: Fill-in-the-Blank (3 Questions)
	38. React uses a DOM to improve performance by minimizing actual DOM changes.
	39. The useEffect hook takes two arguments: a function and an optional array.
	40. The core principle of React data management is called data flow.

Answer Key – React Key Concepts Exam

Section A: Multiple Choice

- 1. **C** To build user interfaces efficiently using components
- 2. **B** A reusable piece of UI logic and structure
- 3. **A** An HTML-like syntax used within JavaScript
- 4. **B** Wrap them in a parent element or use fragments
- 5. **C** useState
- 6. **C** Minimizing direct DOM manipulation by updating only what changed
- 7. **C** It schedules a re-render with the new state
- 8. **B** They help React identify which items have changed
- 9. **C** useEffect
- 10. **C** Props are read-only; state is managed locally

Section B: True/False

- 11. False
- 12. False
- 13. **True**
- 14. False
- 15. **False**

Section C: Short Answer

- 16. JSX is a syntax extension for JavaScript that looks like HTML and is used with React to describe the UI.
- 17. Lifting state up means moving state to a common ancestor to share it among multiple components.
- 18. Because direct mutation doesn't trigger re-renders; always use the setter function.

- 19. When you want the effect to run only once after the first render.
- 20. Fragments let you group elements without adding extra nodes to the DOM.
- 21. Props allow data to be passed into components, making them customizable and reusable.
- 22. It ensures changes are predictable and traceable, which is important for performance.
- 23. Helps React identify which items changed, are added, or are removed.
- 24. Data flows from parent to child components only, not the other way around.
- 25. It schedules an update to the component's state and causes a re-render.

Section D: Code Analysis

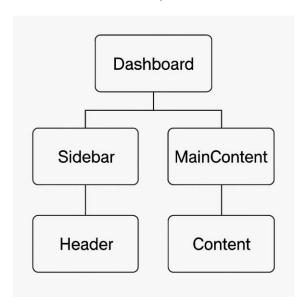
- 26. <h1>Hello World</h1> will render.
- 27. A paragraph with count and a button that increments it.
- 28. Missing parentheses in useState; should be useState(") or similar.
- 29. It will log "Hi" on every render because no dependency array was provided.
- 30. Missing key prop; should use key={item}>.

Section E: Critical Thinking / Extended Response

- 31. Function components are simpler and use hooks; class components use lifecycle methods.
- 32. When two child components need access to shared state, it is lifted to the parent.
- 33. Because useEffect mimics lifecycle methods like componentDidMount.
- 34. To avoid recalculating values or functions unless dependencies change.
- 35. Use cleanup function inside use Effect or Abort Controller to cancel async ops.

Section F: Drawing/Sketch

36. Student drawing: Component tree (e.g., Dashboard → Sidebar, MainContent → Header and Content)



37. Student drawing: One-way data flow of props; upward arrows for event callbacks

Section G: Fill-in-the-Blank

- 38. virtual
- 39. callback, dependency
- 40. one-way