Below are 25 multiple-choice questions. Choose the **best** answer after reading all the options:

1. **How can data be passed between components in React?**
   1. Using props
   2. Using state
   3. Both A and B
   4. None of the above
2. **What is the purpose of the useEffect hook in React?**
   1. Managing component state
   2. Handling side effects
   3. Handling form submissions
   4. Used for memoization purpose
3. **Why is cleanup necessary in the useEffect hook?**
   1. To optimize performance
   2. To prevent memory leaks
   3. To improve readability
   4. Cleanup is not necessary
4. **What are controlled components in React?**
   1. Components with strict access controls
   2. Components managing its’s states using a global state management tool
   3. Components with state and DOM elements holding/referencing the same value as in the state, at all times
   4. Components with conditional rendering
5. **What is the default behavior of forms in HTML, and how do you handle it in React?**
   1. Default behavior: Automatic submission; React: No change required
   2. Default behavior: Automatic submission; React: Prevent default to avoid page reload
   3. Default behavior: No automatic submission; React: Prevent default for optimization
   4. Default behavior: No automatic submission; React: No change required
6. **Which library is commonly used for client-side routing and pages navigation in React?**
   1. React Navigation
   2. React Router
   3. Router.js
   4. Navigation.js
7. **What is the purpose of the fetch API in React?**
   1. To fetch components
   2. To fetch data from a remote server
   3. To fetch images
   4. To fetch state
8. **How can you handle loading states and error responses in asynchronous operations?**
   1. Using try-catch blocks
   2. Using loading and error props
   3. Using loading and error states
   4. Both B and C
   5. To handle loading states, simply use the 'loading=true' attribute on any HTML body tag. For errors, you can print them in the console.
   6. Loading states can be effortlessly managed by relying on React's innate capabilities; no explicit implementation is required. The framework seamlessly addresses loading and error scenarios.
9. **What is the Context API used for in React?**
   1. Handling form states
   2. Managing global state that can be shared by different components throughout the app
   3. Performing asynchronous state updates
   4. Globally managing shared states of the top-level components in the hierarchy
   5. Managing states of low-level components (child/leaf nodes) in the hierarchy tree
10. **Which of the following is a library/tool used for state management in React?**
    1. Context API
    2. React Global
    3. State Manager React
    4. Redux
    5. Both A and B
    6. Both A and D
    7. Both C and D
    8. A, B and D
11. **What is the purpose of using a state management tool?**
    1. Code splitting – so that a lot of states been made within a component do not compromise on readability of the component
    2. To manage states that hold values fetched using asynchronous operations (APIs)
    3. To simplify and improve component rendering process based on state changes
    4. To be able to share state(s) across all the components in your application
    5. To avoid creating so many states in React in the first place
12. **What qualifies as a Single-Page Application (SPA) in React?**
    1. An app with at-least one controlled component
    2. An app that contains just one web page and does not redirect to any other page
    3. An app with a single HTML page that dynamically updates the content of the page
    4. An app with multiple HTML pages that do not redirect to each other – each page is a separate entity
    5. An app that can fit into a single view (width and height) on any device and is not scrollable
13. **What makes React apps quite performant?**
    1. The reuse of components, hence less code is actually shipped in a production build
    2. Reconciliation process through which React updates the Virtual DOM first and then uses the diffing algorithm to make efficient and optimized updates in the Real DOM
    3. Global state management that reduces the state updates on different components down the hierarchy, hence it is so fast
    4. Compatibility with modern browsers
14. **What are yarn and npm used for?**
    1. Both of these are important for starting any JS based application on local server
    2. They help with hot reloading (fast refresh) of React code when some changes are made
    3. Both are package managers used for installing packages and dependencies
    4. They are used for maintaining specific versions of installed packages to maintain consistency throughout the project
15. **What does the useState hook return in React?**
    1. The current value of the state
    2. The previous and current (new) value of the state
    3. The value if it is a primitive data type and reference to the memory if it is non-primitive
    4. An array containing the current state and its updater function
    5. The object that contains a component class instance from which you can access state
16. **What would cause a component to get stuck in a loop of infinite re-rendering?**
    1. If you set a state from within the useEffect hook
    2. If you do not pass a default value while initializing a state
    3. If you set a state from within a useEffect hook, unconditionally
    4. If you set a state from within the useEffect hook with an empty dependency array
    5. If you set a state from within the useEffect hook without a dependency array
    6. When the component is removed from the DOM
17. **What is the purpose of the useContext hook in React?**
    1. Managing side effects
    2. Bringing multiple contexts into one as a single exported module
    3. Consuming shared values and functions from a context in functional components
    4. Implementing states based on conditional rendering
    5. Controlling the lifecycle of top-level components in the application’s hierarchy
18. **What does client-side routing in React application provide?**
    1. Improving server performance
    2. Enhancing user experience by updating UI without full-page reloads
    3. Automatically redirecting user to a custom-built 404 Not Found page if user enters an invalid url path
    4. Both B and C
    5. All of the above
19. **Suppose, in the topbar (menubar/header) of your React app, you need to display a button that says ‘Login’. Now if the user has logged in, you need to conditionally render (switch it to) the user’s name and avatar. What would be the best way to achieve this functionality?**
    1. Using a built-in isAuth() function to return the relevant component to be displayed
    2. Passing a dependency array in the useEffect hook, with a custom isAuthenticated function as a dependency
    3. Leveraging React Router to display a different screen based on the authentication status provided
    4. Checking authentication status in the component's JSX and returning component based on the condition
20. **This is a way to display a button in React Native?**
    1. Button
    2. Touchable opacity
    3. Pressable
    4. Both A & B
    5. Both A & C
    6. Both B & C
    7. None of the above
21. **In React, how can you implement conditional rendering based on authentication status?**
    1. Using a built-in isAuth() function
    2. Passing a dependency array in the useEffect hook
    3. Leveraging React Router to display screen based on the authentication status provided
    4. Checking authentication status in the component's JSX and returning component based on the condition
22. **How do you include images in a React Native project?**
    1. Using the <img> tag
    2. Using the <image> tag
    3. Using the <Image> component
    4. Using the <Picture> component
    5. Using the includeImage() function
23. **What is used for wrapping of multiple elements into a container in React Nativ**e**?**
    1. A container class
    2. A View component
    3. A Container component
    4. A div element
24. **Which one of these is true for map and filter functions in JS?**
    1. Both are array helper methods that iterate through each item in the array but map function returns an array of items returned in each iteration whereas filter function just returns an array of the items that satisfy the condition
    2. The map function is used to loop through items in JSX whereas filter function can be used inside an arrow function only
    3. A map function may or may not return something (based on the implementation) whereas the filter function returns the items that satisfy the provided condition
    4. None of the above
25. **What is true about non-primitive data types in JS?**
    1. Arrays, strings and objects are all non-primitive
    2. Values like number, boolean, string, etc. are non-primitive
    3. Data types that hold the actual values
    4. Data types that hold the references to the memory that hold the actual values
    5. Data types that hold functions are called non-primitives
    6. None of the above