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

1. **What is the purpose of the useState hook in React?**
   1. To manage side effects
   2. To manage component state
   3. To handle routing
   4. To perform data fetching
2. **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
3. **What is the purpose of the useEffect hook in React?**
   1. Managing component state
   2. Handling side effects
   3. Handling form submissions
   4. Controlling component lifecycle
4. **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
5. **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
   4. Components with conditional rendering
6. **In React, how can you handle form input changes?**
   1. Using Redux
   2. Using the useEffect hook
   3. Using controlled components
   4. Using props
7. **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
   2. Default behavior: Automatic submission; React: Prevent default
   3. Default behavior: No automatic submission; React: Prevent default
   4. Default behavior: No automatic submission; React: No change
8. **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
9. **How do you handle authentication and authorization in API calls in React?**
   1. Using localStorage
   2. Using cookies
   3. Sending credentials in headers
   4. All of the above
10. **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
11. **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.
12. **What is the Context API used for in React?**
    1. Handling form states
    2. Managing global state
    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
13. **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
14. **What is the role of Redux?**
    1. Handling side effects
    2. Managing global state
    3. Handling form submissions
    4. Handling component lifecycle
    5. Debugging states and props values in applications easily
15. **What is the purpose of using a state management tool?**
    1. Code splitting – so that a lot of states within the 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 using so many states in React in the first place
16. **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
17. **How does React Router help in creating a Multi-Page React app?**
    1. By automatically generating pages
    2. By allowing the creation of multiple HTML files
    3. By dynamically updating components based on routes
    4. By using iframes for each page
18. **What is the primary advantage of using JSX in React?**
    1. Improved performance
    2. Cleaner syntax
    3. Reduced development time
    4. Compatibility with all browsers
19. **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 dependencies and packages
    4. They are used for maintaining specific versions of installed packages to maintain consistency throughout the project
20. **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
21. **What is the purpose of keys in React when rendering a list of components?**
    1. To improve performance and enforce security in the rendered components
    2. To uniquely identify elements, track changes and efficiently update just the relevant elements in the DOM instead of re-painting the whole list in the DOM
    3. To control the order of rendering elements
    4. To be able to access the component using that key as a unique identifier
22. **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
23. **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/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
24. **How does React handle forms in terms of data binding?**
    1. One-way binding
    2. Two-way binding
    3. Three-way binding (React Component, Global State and DOM)
    4. No binding is involved in React forms
    5. None of the above
25. **What is the primary purpose of client-side routing in a React application?**
    1. Improving server performance
    2. Enhancing user experience by updating UI without full-page reloads
    3. Automatically redirecting user to a 404 Not Found page if user enters an invalid url path
    4. Both B and C
    5. All of the above
26. **What is the main advantage of using the context API over props drilling in React?**
    1. Improved performance by avoiding passing props every time
    2. Simplified data passing between components using a shared state (source of truth)
    3. Better code organization and readability
    4. None of the above
27. **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
28. **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
29. **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
30. **How do you include images in a React Native project?**
    1. Using the img tag
    2. Using the Image component
    3. Using the Picture element
    4. Cleaner syntax
    5. Using the includeImage function
31. **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
32. **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. A map is used to loop through items in JSX whereas filter can be used inside a function
    3. A map function may or may not return something whereas the filter function returns the items that satisfy the provided condition
    4. None of the above
33. **What is true about non-primitive data types?**
    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
34. **Which of the following is a primitive data type?**
    1. Boolean
    2. Object
    3. Number
    4. String
    5. Both A and C
    6. Both B and D
35. **What does an empty array (an array with length 0) evaluate to in JS?**
    1. True
    2. False
    3. undefined
    4. null
36. **What is the output of the following code? {} === {}**
    1. True
    2. Could be true if both hold same references to the memory
    3. False for sure
    4. Both A and C
    5. Both B and D