**Beginner level:**

1. What are React hooks and how do they simplify the process of managing state and lifecycle methods in functional components.
2. The useState() hook and how to use it to manage state in a functional component.
3. The useEffect() hook and how to use it to manage side effects in a functional component.
4. The useContext() hook and how to use it to consume data from a React context in a functional component.
5. Using the useRef() hook to create a reference to a DOM element in a functional component.

**Intermediate level:**

1. Best practices for using React hooks in a React application.
2. Using the useReducer() hook to manage complex state in a functional component.
3. Using the useMemo() and useCallback() hooks to optimize performance in a functional component.
4. Using the useLayoutEffect() hook to perform actions after DOM rendering but before painting.
5. Creating custom hooks to encapsulate stateful logic and share it between components.

**Advanced level:**

1. Using multiple state hooks to manage complex state in a functional component.
2. Using React hooks with higher-order components (HOC) to manage state and lifecycle methods in a React application.
3. Understanding the performance impact of using React hooks and optimizing their use in a React application.
4. Using React hooks with Redux or other state management libraries to manage the state of a React application.
5. Using TypeScript with React hooks to improve the type safety of a React application.
6. Understanding the internals of the React framework and how hooks are processed by the React engine.