1. What is React? Answer: React is a JavaScript library developed by Facebook for building user interfaces, particularly for single-page applications. It allows developers to create reusable UI components.

---

2. What are the key features of React? Answer:

- JSX (JavaScript XML)
- Virtual DOM
- Component-based architecture
- Unidirectional data flow
- Lifecycle methods

---

3. What is JSX? Answer: JSX stands for JavaScript XML. It allows writing HTML elements in JavaScript and placing them in the DOM without using functions like createElement().

---

4. What is the virtual DOM? Answer: The virtual DOM is a lightweight JavaScript object that is a copy of the real DOM. React uses it to optimize rendering by updating only the changed parts of the UI.

---

5. What are components in React? Answer: Components are the building blocks of a React application. They can be functional or class-based and can manage their own state and props.

---

6. What is the difference between state and props? Answer:

- **Props**: Read-only, passed from parent to child.
- **State**: Managed within the component, mutable, and used to track data that changes over time.

---

7. What are React Hooks? Answer: React Hooks are functions that let you use state and other React features in functional components. Common hooks include useState, useEffect, and useContext.

---

8. What is the useEffect hook? Answer: useEffect performs side effects in functional components, like data fetching, subscriptions, or manually changing the DOM.

---

9. What is lifting state up in React? Answer: Lifting state up refers to moving the state from child components to their common parent to share the state among those children.

---

10. What are keys in React and why are they important? Answer: Keys help React identify which items have changed, are added, or are removed. They should be unique to each element in an array.

---

11. What is the difference between a controlled and uncontrolled component? Answer:

- **Controlled**: Form data is handled by a React component using state.
- **Uncontrolled**: Form data is handled by the DOM directly using refs.

---

12. How does React handle events? Answer: React uses synthetic events, which are wrappers around the browser's native events. Events are handled using camelCase syntax and JavaScript functions.

---

13. What is Redux? Answer: Redux is a state management library for JavaScript applications. It is commonly used with React to manage the application's global state.

---

14. What are the main principles of Redux? Answer:

- Single source of truth
- State is read-only
- Changes are made with pure functions (reducers)

---

15. How do you optimize performance in a React application? Answer:

- Use React.memo
- Lazy loading components
- Code splitting
- Avoid inline functions in render
- Use useCallback and useMemo

---

16. What is React Router? Answer: React Router is a standard routing library for React. It enables the navigation among views and allows for dynamic route matching.

---

17. What is the difference between useMemo and useCallback? Answer:

- **useMemo**: Memoizes the result of a computation.
- **useCallback**: Memoizes the function definition itself.

---

18. What is context in React? Answer: Context provides a way to pass data through the component tree without having to pass props down manually at every level.

---

19. What are higher-order components (HOCs)? Answer: HOCs are functions that take a component and return a new component, often used for reusing component logic.

---

20. How does React differ from other frameworks like Angular or Vue? Answer: React focuses only on the view layer and uses a component-based approach, while Angular is a full-fledged MVC framework, and Vue is similar to React but has its own syntax and ecosystem.
