1. **ES6+ Features:**
   * Arrow functions
   * Template literals
   * Destructuring assignment
   * let and const
   * Spread/rest operators
   * Classes and inheritance
   * Promises and async/await
   * Modules (import/export)
2. **Functional Programming:**
   * Higher-order functions
   * Pure functions
   * Immutability
   * Function composition
3. **React Basics:**
   * JSX (JavaScript XML)
   * Components and Props
   * State and Lifecycle
   * Handling events
   * Conditional rendering
   * Lists and keys
4. **State Management:**
   * State in class components
   * useState hook for functional components
   * State lifting and prop drilling
   * Context API for global state
5. **Hooks:**
   * useState
   * useEffect
   * useContext
   * useReducer
   * Custom hooks
6. **React Router:**
   * Navigation in React applications
   * Route parameters and query strings
7. **Redux (Optional):**
   * Actions, reducers, and the store
   * Connecting React with Redux
   * Async operations with middleware (thunk, saga)
8. **API Calls:**
   * Fetch API or Axios for making HTTP requests
   * Handling asynchronous operations
9. **Forms in React:**
   * Controlled components
   * Form handling and validation
10. **Responsive Design:**
    * Media queries
    * Flexbox and Grid layout
11. **Testing:**
    * Unit testing with Jest and React Testing Library
    * Snapshot testing
12. **Build Tools:**
    * npm or Yarn for package management
    * Webpack for bundling
    * Babel for transpiling
13. **Version Control:**
    * Git and GitHub/GitLab/Bitbucket for version control
14. **Modern JavaScript Tooling:**
    * ESLint for code linting
    * Prettier for code formatting
15. **Browser Developer Tools:**
    * Understanding how to use browser developer tools for debugging
16. **Basic Understanding of CSS:**
    * Styling components
    * CSS-in-JS libraries (e.g., styled-components)
17. **Error Handling:**
    * Handling errors in React applications
18. **Security Best Practices:**
    * Understanding security concerns related to client-side applications