# React JS Interview Preparation Topics

## 1. Core React Concepts

1. Components

- Functional Components

- Definition and syntax

- Advantages over class components

- Class Components

- Definition and syntax

- Lifecycle methods (e.g., componentDidMount, componentDidUpdate, componentWillUnmount)

- Higher-Order Components (HOCs)

- Definition and use cases

- Presentational vs. Container Components

- Understanding the differences and when to use each type

2. JSX

- Syntax

- Writing JSX expressions

- Differences between HTML and JSX

- Transformations

- How JSX transforms to JavaScript using Babel

3. State and Props

- State Management

- Using useState in functional components

- State vs. props: differences and use cases

- Props

- Passing props to components

- PropTypes for type-checking props

4. Event Handling

- Handling Events

- Syntax for event handling (e.g., onClick, onChange)

- Event object and synthetic events

- Form Handling

- Controlled vs. uncontrolled components

- Form submission handling

5. Conditional Rendering

- Methods of Conditional Rendering

- Using ternary operators

- Using logical AND (&&) operator

- Switch statement for multiple conditions

6. Lists and Keys

- Rendering Lists

- Using map() to render lists

- Keys in Lists

- Importance of keys for performance

- Best practices for using keys

## 2. Hooks

1. Basic Hooks

- useState

- Managing state in functional components

- Initial state and updating state

- useEffect

- Managing side effects

- Dependency arrays and cleanup functions

2. Custom Hooks

- Creating Custom Hooks

- Definition and use cases

- Examples of reusable logic with custom hooks

## 3. Routing and State Management

1. React Router

- Basic Setup

- Installing and configuring React Router

- Using BrowserRouter, Route, and Link

- Nested Routes

- Understanding nested routes and their implementation

- Redirects and Route Parameters

- Handling redirects and dynamic route parameters

2. Context API

- Creating Context

- Using createContext and useContext

- Provider and consumer components

- Global State Management

- Managing global state with the Context API

3. Redux

- Introduction to Redux

- Understanding actions, reducers, and the store

- Using Redux with React

- Connecting React components to Redux using connect and useSelector

- Middleware and Asynchronous Actions

- Using Redux Thunk or Redux Saga for async operations

## 4. Performance Optimization

1. Memoization

- React.memo

- Preventing unnecessary re-renders of components

- useMemo and useCallback

- Optimizing expensive calculations and callback functions

2. Code Splitting

- React.lazy and Suspense

- Implementing lazy loading for components

- Using Suspense for fallback UI

## 5. Testing

1. Unit Testing

- Jest

- Writing and running tests with Jest

- React Testing Library

- Testing React components and user interactions

2. Integration Testing

- Testing Component Interactions

- Writing tests for component integrations and state changes

## 6. Advanced Topics

1. TypeScript with React

- Introduction to TypeScript

- Benefits of using TypeScript in React projects

- Typing Props and State

- Defining types for component props and state

2. Server-Side Rendering (SSR) and Static Site Generation (SSG)

- Next.js

- Setting up a Next.js project

- Understanding SSR and SSG concepts

3. Progressive Web Apps (PWAs)

- Building PWAs with React

- Understanding service workers and caching

4. React Native (if relevant)

- Introduction to React Native

- Differences between React and React Native

- Building mobile applications with React Native

## 7. Additional Concepts

1. Error Boundaries

- Implementing Error Boundaries

- Catching errors in child components and rendering fallback UI

2. React DevTools

- Using React DevTools

- Debugging and optimizing React applications with DevTools

3. Design Patterns

- Common Design Patterns in React

- Render props, compound components, and other patterns