1. Virtual DOM Concepts

Direct MCQs:

1. What is the Virtual DOM in React?
   a) A copy of the real DOM in memory
   b) A server-side representation of the DOM
   c) A third-party library
   d) A template engine
   Answer: a) A copy of the real DOM in memory

2. Why does React use a Virtual DOM?
   a) To increase server load
   b) To improve performance by minimizing direct DOM manipulations
   c) To replace HTML
   d) To manage CSS styles
   Answer: b) To improve performance by minimizing direct DOM manipulations

3. How does React update the actual DOM efficiently?
   a) Directly renders everything
   b) Uses a diffing algorithm to update only changed elements
   c) Clears the DOM and re-renders
   d) Uses AJAX calls
   Answer: b) Uses a diffing algorithm to update only changed elements

Scenario-Based MCQs:
4. A developer updates a component state multiple times rapidly. How does React handle DOM updates?
a) Updates DOM on every state change
b) Updates Virtual DOM and applies minimal changes to real DOM
c) Ignores updates
d) Reloads the entire page
Answer: b) Updates Virtual DOM and applies minimal changes to real DOM

5. In a React app, a list of 1000 items changes only in one element. How is performance optimized?
   a) Re-renders all 1000 elements
   b) Virtual DOM updates only the changed element in real DOM
   c) Uses global variables to track changes
   d) Requires manual DOM updates
   Answer: b) Virtual DOM updates only the changed element in real DOM

2. JSX Concepts

Direct MCQs:

1. JSX stands for:
   a) JavaScript XML
   b) Java Standard XML
   c) JavaScript XHTML

d) JavaScript Export
Answer: a) JavaScript XML

2. Which of the following is correct JSX syntax to display text "Hello"?
   a) `<h1>Hello</h1>`
   b) `h1("Hello")`
   c) `{Hello}`
   d) `["Hello"]`
   Answer: a) `<h1>Hello</h1>`

3. In JSX, how do you embed JavaScript expressions inside HTML?
   a) `<h1>JavaScript</h1>`
   b) `{expression}`
   c) `[expression]`
   d) `(expression)`
   Answer: b) `{expression}`

Scenario-Based MCQs:
4. A developer wants to display a variable `name` inside a paragraph. Which JSX is correct?
a) `<p>name</p>`
b) `<p>{name}</p>`
c) `<p>${name}</p>`
d) `<p>[name]</p>`
Answer: b) `<p>{name}</p>`

5. A developer tries to return two sibling elements without a wrapper. What is the correct JSX solution?
   a) Wrap them inside a `<div>` or `<> </>` fragment
   b) Use multiple return statements
   c) Use `document.createElement()`
   d) Use an array only
   Answer: a) Wrap them inside a `<div>` or `<> </>` fragment

3. React Project Generation and Build Tool

Direct MCQs:

1. Which command is commonly used to create a React project?
   a) `npm start react-app`
   b) `npx create-react-app my-app`
   c) `npm react create`
   d) `react-init my-app`
   Answer: b) `npx create-react-app my-app`

2. Which build tool does Create React App use internally?
   a) Webpack
   b) Gulp

c) Grunt
d) Parcel
Answer: a) Webpack

3. Which command runs a React project in development mode?
   a) `npm build`
   b) `npm start`
   c) `npm run server`
   d) `npm serve`
   Answer: b) `npm start`

Scenario-Based MCQs:
4. A developer wants to create a production-ready build for deployment. Which command is used?
a) `npm start`
b) `npm run build`
c) `npm dev`
d) `npm serve`
Answer: b) `npm run build`

5. A developer wants to generate a new React project using Yarn instead of NPM. Which command is correct?
   a) `yarn create react-app my-app`
   b) `yarn react init my-app`
   c) `yarn npm create react-app`
   d) `yarn build react-app`
   Answer: a) `yarn create react-app my-app`

4. Structure of the React Application

Direct MCQs:

1. Which folder contains the main component files in a standard React app?
   a) `/public`
   b) `/src`
   c) `/build`
   d) `/node_modules`
   Answer: b) `/src`

2. Which file serves as the entry point of a React app?
   a) `App.js`
   b) `index.js`
   c) `package.json`
   d) `main.js`
   Answer: b) `index.js`

3. Where do static assets like images go in a standard React project?
   a) `/src`

b) `/public`
c) `/assets`
d) `/build`
Answer: b) `/public`

Scenario-Based MCQs:
4. A developer wants to import a component `Header.js` in `App.js`. Where should `Header.js` ideally reside?
a) `/src/components/Header.js`
b) `/public/Header.js`
c) `/build/Header.js`
d) `/node_modules/Header.js`
Answer: a) `/src/components/Header.js`

5. A developer modifies `App.js` but sees no changes on the browser. Which folder is likely responsible for served files?
a) `/src`
b) `/public`
c) `/build`
d) `/node_modules`
Answer: c) `/build`

5. Class and Functional Components

Direct MCQs:

1. Which is the correct way to define a functional component?
a) `function MyComponent(){ return <h1>Hello</h1> }`
b) `class MyComponent extends React.Component`
c) `var MyComponent = new Component()`
d) `component MyComponent(){}`
Answer: a) `function MyComponent(){ return <h1>Hello</h1> }`

2. Which method is used to render JSX in a class component?
a) `render()`
b) `return()`
c) `display()`
d) `show()`
Answer: a) `render()`

3. Which hook is used for state management in functional components?
a) `useState()`
b) `useEffect()`
c) `useReducer()`
d) `useContext()`
Answer: a) `useState()`

Scenario-Based MCQs:
4. A developer wants a component to display current time that updates every second. Which component type is suitable with hooks?
a) Functional component with `useState` and `useEffect`
b) Class component only
c) Functional component without hooks
d) Pure CSS component
Answer: a) Functional component with `useState` and `useEffect`

5. A developer wants a component to log a message once when it mounts. Which lifecycle feature is suitable?
   a) `componentDidMount()` in class component or `useEffect(()=>{},[])` in functional component
   b) `render()`
   c) `useState()`
   d) `componentWillUnmount()`
   Answer: a) `componentDidMount()` in class component or `useEffect(()=>{},[])` in functional component

6. Event Handling

Direct MCQs:

1. How do you attach a click event in React JSX?
   a) `<button onClick={handleClick}>Click</button>`
   b) `<button onclick="handleClick()">Click</button>`
   c) `<button click={handleClick}>Click</button>`
   d) `<button eventClick={handleClick}>Click</button>`
   Answer: a) `<button onClick={handleClick}>Click</button>`

2. How do you prevent default behavior in React events?
   a) `event.preventDefault()`
   b) `return false`
   c) `stopDefault()`
   d) `prevent()`
   Answer: a) `event.preventDefault()`

3. Which event is used to detect changes in an input field?
   a) `onChange`
   b) `onClick`
   c) `onHover`
   d) `onSelect`
   Answer: a) `onChange`

Scenario-Based MCQs:
4. A developer wants to log input text as the user types. Which event and method are correct?
a) `onChange={e => console.log(e.target.value)}`

b) `onClick={console.log(input.value)}`
c) `onHover={console.log(input.value)}`
d) `onSelect={console.log(input.value)}`
Answer: a) `onChange={e => console.log(e.target.value)}`

5. A form should not reload the page when submitted. Which event handling is required?
   a) `onSubmit={e => e.preventDefault()}`
   b) `onClick`
   c) `onChange`
   d) `onLoad`
   Answer: a) `onSubmit={e => e.preventDefault()}`

1. Controlled Components

Direct MCQs:

1. What is a controlled component in React?
   a) Component with internal state only
   b) Component whose form data is handled by React state
   c) Component using default HTML form behavior
   d) Component that cannot receive props
   Answer: b) Component whose form data is handled by React state

2. Which hook is commonly used to manage controlled components?
   a) `useEffect()`
   b) `useState()`
   c) `useRef()`
   d) `useContext()`
   Answer: b) `useState()`

3. How do you update the state of a controlled input?
   a) Using `event.preventDefault()`
   b) Using `setState` or `useState` setter function in `onChange`
   c) Using `document.getElementById()`
   d) Using `defaultValue` only
   Answer: b) Using `setState` or `useState` setter function in `onChange`

Scenario-Based MCQs:
4. A form input must reflect the component's state as the user types. Which approach should be used?
a) Controlled component
b) Uncontrolled component
c) Static HTML input
d) Read-only input
Answer: a) Controlled component

5. A developer wants a text input field where the value is always in sync with a `name` state variable. Which JSX is correct?
   a) `<input value={name} onChange={e => setName(e.target.value)} />`
   b) `<input defaultValue={name} />`
   c) `<input value={name} />`
   d) `<input ref={name} />`
   Answer: a) `<input value={name} onChange={e => setName(e.target.value)} />`

2. Passing Arguments Between Components

Direct MCQs:

1. How are data values passed from a parent to a child component?
   a) Using `props`
   b) Using `state`
   c) Using `ref`
   d) Using `context` only
   Answer: a) Using `props`

2. How does a child component send data to its parent?
   a) Using a function passed via props
   b) Using local variables
   c) Using `defaultProps`
   d) Using JSX only
   Answer: a) Using a function passed via props

3. Which of the following is valid for passing a numeric prop named `age` to a child component?
   a) `<Child age={25} />`
   b) `<Child age="25" />`
   c) `<Child age={“25”} />`
   d) `<Child age(25) />`
   Answer: a) `<Child age={25} />`

Scenario-Based MCQs:
4. A parent component wants to increment a counter in the child component. How should it pass the increment function?
a) As a prop function: `<Child increment={handleIncrement} />`
b) As a state variable only
c) Using defaultValue
d) Using uncontrolled ref
Answer: a) As a prop function: `<Child increment={handleIncrement} />`

5. A child component needs to display a list received from the parent. How should it access the data?

a) `props.list`
b) `state.list`
c) `useEffect()`
d) `useRef()`
Answer: a) `props.list`

3. Default Property Initialization

Direct MCQs:

1. How do you assign default props to a React component?
   a) `Component.defaultProps = {}`
   b) `Component.defaultState = {}`
   c) `Component.props.default = {}`
   d) `Component.initializeProps = {}`
   Answer: a) `Component.defaultProps = {}`

2. Why are default props used?
   a) To prevent errors when a prop is not passed
   b) To update state automatically
   c) To render DOM manually
   d) To manage lifecycle
   Answer: a) To prevent errors when a prop is not passed

3. Can default props be used with functional components?
   a) Yes
   b) No
   c) Only with class components
   d) Only with JSX
   Answer: a) Yes

Scenario-Based MCQs:
4. A component has a `title` prop, but sometimes the parent does not pass it. How should the component ensure a fallback value?
a) Use `Component.defaultProps = { title: "Default Title" }`
b) Use `state.title` only
c) Use `ref.title`
d) Use `onClick`
Answer: a) Use `Component.defaultProps = { title: "Default Title" }`

5. A developer wants a functional component to display `"Guest"` if no `name` prop is provided. What's the correct way?
   a) `Component.defaultProps = { name: "Guest" }`
   b) `const name = "Guest"`
   c) `props.name = "Guest"`
   d) Use `useState("Guest")`
   Answer: a) `Component.defaultProps = { name: "Guest" }`

4. PropTypes

Direct MCQs:

1. What is PropTypes used for in React?
   a) Type checking of props
   b) State management
   c) DOM manipulation
   d) Handling events
   Answer: a) Type checking of props

2. Which package is used for PropTypes in modern React?
   a) `prop-types`
   b) `react-types`
   c) `react-propcheck`
   d) `propcheck`
   Answer: a) `prop-types`

3. How do you define a required string prop named `title`?
   a) `Component.propTypes = { title: PropTypes.string.isRequired }`
   b) `Component.defaultProps = { title: string }`
   c) `Component.state = { title: string }`
   d) `Component.propTypes = { title: string.required }`
   Answer: a) `Component.propTypes = { title: PropTypes.string.isRequired }`

Scenario-Based MCQs:
4. A parent component passes a number to a prop expecting a string. What does PropTypes do?
a) Throws a console warning in development
b) Crashes the app
c) Converts number to string automatically
d) Ignores the prop
Answer: a) Throws a console warning in development

5. A component has an optional boolean prop `isActive`. How should PropTypes be defined?
   a) `isActive: PropTypes.bool`
   b) `isActive: PropTypes.bool.isRequired`
   c) `isActive: PropTypes.boolean`
   d) `isActive: PropTypes.optionalBool`
   Answer: a) `isActive: PropTypes.bool`

5. Uncontrolled Components

Direct MCQs:

1. What is an uncontrolled component in React?
   a) Component whose form data is handled by DOM, not React state
   b) Component with no props
   c) Component using defaultProps only
   d) Component with hooks only
   Answer: a) Component whose form data is handled by DOM, not React state

2. Which hook is commonly used to access DOM values in uncontrolled components?
   a) `useRef()`
   b) `useState()`
   c) `useEffect()`
   d) `useCallback()`
   Answer: a) `useRef()`

3. Which attribute is commonly used to set initial value in uncontrolled components?
   a) `defaultValue`
   b) `value`
   c) `refValue`
   d) `initialValue`
   Answer: a) `defaultValue`

Scenario-Based MCQs:
4. A developer wants a form input where the value is retrieved only on form submission. Which approach should be used?
a) Uncontrolled component with `ref`
b) Controlled component
c) useState for every keystroke
d) Event handler only
Answer: a) Uncontrolled component with `ref`

5. A text input should have the initial value "Hello" but not update the state on every keystroke. Which JSX is correct?
   a) `<input defaultValue="Hello" ref={inputRef} />`
   b) `<input value="Hello" onChange={...} />`
   c) `<input ref="Hello" />`
   d) `<input initialValue="Hello" />`
   Answer: a) `<input defaultValue="Hello" ref={inputRef} />`

6. Component Lifecycle Methods

Direct MCQs:

1. Which lifecycle method is called immediately after a component is mounted?
   a) `componentDidMount()`
   b) `componentWillMount()`
   c) `render()`
   d) `getDerivedStateFromProps()`
   Answer: a) `componentDidMount()`

2. Which lifecycle method is invoked before a component is removed from the DOM?
   a) `componentWillUnmount()`
   b) `componentDidUpdate()`
   c) `getSnapshotBeforeUpdate()`
   d) `shouldComponentUpdate()`
   Answer: a) `componentWillUnmount()`

3. Which lifecycle method is used to update state in response to prop changes?
   a) `getDerivedStateFromProps()`
   b) `componentDidMount()`
   c) `render()`
   d) `componentWillUnmount()`
   Answer: a) `getDerivedStateFromProps()`

Scenario-Based MCQs:
4. A component fetches data from an API when mounted. Which lifecycle method is ideal?
a) `componentDidMount()`
b) `render()`
c) `componentWillUnmount()`
d) `shouldComponentUpdate()`
Answer: a) `componentDidMount()`

5. A component needs to clean up timers before being destroyed. Which lifecycle method is appropriate?
   a) `componentWillUnmount()`
   b) `componentDidMount()`
   c) `getSnapshotBeforeUpdate()`
   d) `componentDidUpdate()`
   Answer: a) `componentWillUnmount()`

1. Multiple Components Interaction & Backend Data (json-server)

Direct MCQs:

1. How can data be shared between parent and child components?
   a) Using props
   b) Using CSS
   c) Using external HTML
   d) Using global variables only
   Answer: a) Using props

2. Which React hook is commonly used to fetch data from a backend?
   a) `useEffect()`
   b) `useState()`
   c) `useRef()`

d) `useContext()`
Answer: a) `useEffect()`

3. Which library is used to simulate a backend server for React testing?
   a) `json-server`
   b) `axios-server`
   c) `react-server`
   d) `express-json`
   Answer: a) `json-server`

Scenario-Based MCQs:
4. A developer wants a React app to fetch a list of users from `json-server` when the component mounts. Which approach is correct?
a) Use `useEffect()` to call `fetch("http://localhost:3000/users")` and update state
b) Use `useRef()` only
c) Set state manually without API call
d) Use only `componentDidUpdate()`
Answer: a) Use `useEffect()` to call `fetch("http://localhost:3000/users")` and update state

5. Two sibling components need access to the same fetched data. What is the recommended approach?
   a) Lift state up to the parent and pass via props
   b) Use separate fetch calls in each component
   c) Use CSS variables
   d) Hardcode the data in both components
   Answer: a) Lift state up to the parent and pass via props

2. Form Handling using Formik & Yup

1. Which library is commonly used for form state management in React?
   a) Formik
   b) React Router
   c) Redux
   d) Axios
   Answer: a) Formik

2. Which library is used for validating forms in React?
   a) Yup
   b) ReactDOM
   c) Lodash
   d) Axios
   Answer: a) Yup

3. Which Formik prop is used to handle form submission?
   a) `onSubmit`
   b) `handleChange`

c) `initialValues`
d) `validationSchema`
Answer: a) `onSubmit`

Scenario-Based MCQs:
4. A developer wants to validate an email field in a Formik form. Which combination is correct?
a) Use Formik for form state and Yup for email validation
b) Use only HTML5 validation
c) Use Redux for validation
d) Use CSS for validation
Answer: a) Use Formik for form state and Yup for email validation

5. A form should reset its values after successful submission. Which Formik method is suitable?
a) `resetForm()` inside `onSubmit`
b) `setState()`
c) `useRef()`
d) `componentWillUnmount()`
Answer: a) `resetForm()` inside `onSubmit`

3. Higher Order Components (HOC)

1. What is a Higher Order Component (HOC) in React?
a) A function that takes a component and returns a new component
b) A React component using `useState()`
c) A built-in React hook
d) A router component
Answer: a) A function that takes a component and returns a new component

2. HOCs are used primarily for:
a) Code reuse, logic abstraction, and cross-cutting concerns
b) Styling components
c) Handling events only
d) Managing props type-checking
Answer: a) Code reuse, logic abstraction, and cross-cutting concerns

3. Which of the following is a valid HOC pattern?
a) `const EnhancedComponent = withFeature(WrappedComponent)`
b) `const Component = useState(WrappedComponent)`
c) `const Component = fetch(WrappedComponent)`
d) `const HOC = <Component />`
Answer: a) `const EnhancedComponent = withFeature(WrappedComponent)`

Scenario-Based MCQs:
4. A developer wants to log props for multiple components without repeating code. What should be used?
a) HOC to wrap the components and add logging
b) Add logging inside every component

c) CSS for props
d) JSON-server
Answer: a) HOC to wrap the components and add logging

5. A HOC needs to pass extra props to the wrapped component. How is it done?
   a) `<WrappedComponent {...props} extraProp={value} />`
   b) Modify `state` directly
   c) Change `document.getElementById()`
   d) Use Redux only
   Answer: a) `<WrappedComponent {...props} extraProp={value} />`

4. Context API

Direct MCQs:

1. What is the primary use of React Context API?
   a) Sharing state across multiple components without prop drilling
   b) Managing CSS
   c) Routing between pages
   d) Fetching data
   Answer: a) Sharing state across multiple components without prop drilling

2. Which React method creates a context object?
   a) `React.createContext()`
   b) `useContext()`
   c) `useState()`
   d) `React.Context()`
   Answer: a) `React.createContext()`

3. How does a child component consume context?
   a) Using `useContext(ContextName)`
   b) Using `props` only
   c) Using `state`
   d) Using `ref`
   Answer: a) Using `useContext(ContextName)`

Scenario-Based MCQs:
4. A theme object should be accessible by all components in an app. What's the correct approach?
a) Wrap components inside `ThemeContext.Provider` and pass value
b) Pass theme via props to each component manually
c) Use global CSS only
d) Use Redux only
Answer: a) Wrap components inside `ThemeContext.Provider` and pass value

5. A child component deep in the tree needs access to user authentication info. Which API avoids prop drilling?
   a) Context API
   b) CSS variables
   c) Uncontrolled inputs

d) HOC only
Answer: a) Context API

5. React Router Concepts

Direct MCQs:

1. Which library is used for routing in React?
   a) `react-router-dom`
   b) `react-router-native`
   c) `react-router`
   d) All of the above
   Answer: d) All of the above

2. Which component defines a route in React Router?
   a) `<Route path="/home" element={<Home />} />`
   b) `<Link path="/home" />`
   c) `<Switch path="/home" />`
   d) `<Router path="/home" />`
   Answer: a) `<Route path="/home" element={<Home />} />`

3. Which component wraps all routes in React Router v6?
   a) `<Routes>`
   b) `<Switch>`
   c) `<Router>`
   d) `<Route>`
   Answer: a) `<Routes>`

Scenario-Based MCQs:
4. A developer wants to navigate programmatically after a successful form submission. Which hook is used?
a) `useNavigate()`
b) `useLocation()`
c) `useParams()`
d) `useEffect()`
Answer: a) `useNavigate()`

5. A URL `/profile/:id` should render a component showing user info based on `id`. How to access `id`?
   a) Using `useParams()`
   b) Using `props.id` only
   c) Using `useState()`
   d) Using `context`
   Answer: a) Using `useParams()`