MSc. CA Semester-1

102-01 Web Development and Design

Question Bank

Unit 1:

MCQs

1. **What is the correct syntax to display "Hello World" in JavaScript?**
    - A) document.write("Hello World")
    - B) response.write("Hello World")
    - C) echo "Hello World"
    - D) console.log("Hello World")

Answer: A) document.write("Hello World")

2. **Which of the following is not a valid JavaScript variable name?**
    - A) _name
    - B) $age
    - C) 1name
    - D) firstName

Answer: C) 1name

3. **Which of the following is true about semicolons in JavaScript?**
    - A) Semicolons are mandatory.
    - B) Semicolons are optional but recommended.
    - C) Semicolons are not allowed.
    - D) Semicolons are only used in loops.

Answer: B) Semicolons are optional but recommended.

4. **Which method is used to round a number down to the nearest integer in JavaScript?**
    - A) Math.round()
    - B) Math.ceil()
    - C) Math.floor()
    - D) Math.trunc()

Answer: C) Math.floor()

5. **Which JavaScript method is used to write HTML to the output stream?**
    - A) document.write()
    - B) document.output()
    - C) document.print()
    - D) document.log()

Answer: A) document.write()

6. **Which of the following is not a JavaScript data type?**
    - A) String
    - B) Number
    - C) Boolean
    - D) Character

Answer: D) Character

7. **How do you create a function in JavaScript?**
    - A) function myFunction()
    - B) def myFunction()
    - C) create myFunction()
    - D) function:myFunction()

Answer: A) function myFunction()

8. **Which of the following methods is used to add an event handler to an element in JavaScript?**
    - A) addListener()
    - B) attachEvent()
    - C) addEventListener()
    - D) onEvent()

Answer: C) addEventListener()

9. **Which method is used to create a new HTML element in JavaScript?**
    - A) document.newElement()
    - B) document.createElement()
    - C) document.buildElement()

- D) document.makeElement()

Answer: B) document.createElement()

10. **Which object represents the browser window in JavaScript?**
    - A) screen
    - B) document
    - C) window
    - D) navigator

Answer: C) window

11. **Which method would you use to move a window to a specified position?**
    - A) moveBy()
    - B) moveTo()
    - C) moveWindow()
    - D) relocate()

Answer: B) moveTo()

12. **How do you write a comment in JavaScript?**
    - A) /* comment */
    - B) // comment
    - C) <!-- comment -->
    - D) All of the above

Answer: D) All of the above

13. **What is the correct way to declare an array in JavaScript?**
    - A) var colors = "red", "green", "blue"
    - B) var colors = (1:"red", 2:"green", 3:"blue")
    - C) var colors = ["red", "green", "blue"]
    - D) var colors = {"red", "green", "blue"}

Answer: C) var colors = ["red", "green", "blue"]

14. **Which method is used to access an HTML element by its ID?**
    - A) document.getElementById()
    - B) document.querySelector()

- C) document.getElementsByClassName()
- D) document.getElementsByTagName()

Answer: A) document.getElementById()

15. **Which method is used to convert a string into a number in JavaScript?**
    - A) parseFloat()
    - B) parseInt()
    - C) Number()
    - D) All of the above

Answer: D) All of the above

Short Questions

1. What is the purpose of using the <script> tag in HTML?
2. Explain the difference between var, let, and const in JavaScript.
3. What is the Document Object Model (DOM)?
4. How do you add an event handler to a button click in JavaScript?
5. What is the purpose of Math.floor() method in JavaScript?
6. How do you declare a JavaScript object?
7. What is the use of document.write() in JavaScript?
8. Explain the concept of a JavaScript function.
9. What are the different ways to create a new date object in JavaScript?
10. How do you write a multi-line comment in JavaScript?
11. What is the difference between == and === in JavaScript?
12. What is the purpose of getElementById() method in JavaScript?
13. Explain how the switch statement works in JavaScript.
14. What is the purpose of the window.prompt() method?
15. How do you loop through the properties of an object in JavaScript?

Long Questions

1. Explain the advantages and limitations of using JavaScript for client-side scripting.
2. Discuss the various methods and properties available in the Math object in JavaScript.
3. How does the DOM tree represent an HTML document, and how can JavaScript manipulate it?

4. Describe the difference between for, while, and do-while loops in JavaScript, with examples.
5. How do JavaScript functions work, and what is the difference between function declarations and function expressions?
6. Explain the process of creating, accessing, and modifying objects in JavaScript.
7. Describe the different ways to handle events in JavaScript. Provide examples.
8. How can you create and manipulate date objects in JavaScript? Explain with examples.
9. Discuss the different dialog boxes available in JavaScript and their use cases.
10. Explain the purpose and use of conditional statements in JavaScript. Provide examples of if, else, else if, and switch statements.

Unit-2:

MCQ

1. What is JSX in React?
    - a) A templating language
    - b) A syntax extension for JavaScript
    - c) A programming language
    - d) A styling framework

Answer: b) A syntax extension for JavaScript

2. Which method is used to fetch data in React?
    - a) fetchAPI()
    - b) fetch()
    - c) getData()
    - d) axios()

Answer: b) fetch()

3. Which of the following is used to pass data from a parent component to a child component in React?
    - a) State
    - b) Props
    - c) Event

- d) Context

Answer: b) Props

4. In JSX, how do you embed a JavaScript expression within HTML-like syntax?
    - a) { }
    - b) [ ]
    - c) ( )
    - d) <>

Answer: a) { }

5. Which hook is commonly used for state management in functional components?
    - a) useEffect
    - b) useReducer
    - c) useState
    - d) useContext

Answer: c) useState

6. What is the purpose of the key attribute in list rendering?
    - a) To provide a unique identifier for each list item
    - b) To style the list items
    - c) To bind event listeners to the list items
    - d) To define the data type of list items

Answer: a) To provide a unique identifier for each list item

7. Which React method is used to create portals?
    - a) React.createPortal()
    - b) ReactDOM.createPortal()
    - c) React.createElement()
    - d) ReactDOM.createElement()

Answer: b) ReactDOM.createPortal()

8. How do you prevent the default behavior of an event in React?
    - a) stopPropagation()
    - b) preventDefault()

- c) cancelEvent()
- d) stopDefault()

Answer: b) preventDefault()

9. What does the useEffect hook do?
    - a) Manages state
    - b) Handles side effects
    - c) Binds events
    - d) Updates props

Answer: b) Handles side effects

10. What does fetch() return?
    - a) A Promise
    - b) A Function
    - c) An Object
    - d) A Response

Answer: a) A Promise

11. Which of the following is true about props?
    - a) Props are mutable
    - b) Props are read-only
    - c) Props are not reusable
    - d) Props cannot be passed to child components

Answer: b) Props are read-only

12. In React, state is primarily used to:
    - a) Pass data between components
    - b) Manage dynamic data within a component
    - c) Style components
    - d) Handle HTTP requests

Answer: b) Manage dynamic data within a component

13. Which of the following is used to update the state in class-based components?
    - a) setState

- b) useState
- c) updateState
- d) modifyState

Answer: a) setState

14. What happens if you don't provide a key when rendering lists in React?
    - a) Nothing happens
    - b) React throws an error
    - c) React will give a warning in the console
    - d) The list items will not be rendered

Answer: c) React will give a warning in the console

15. Which method is used to handle HTTP POST requests with fetch()?
    - a) method: 'GET'
    - b) method: 'POST'
    - c) type: 'POST'
    - d) fetchPost()

Answer: b) method: 'POST'

Short Questions

1. What is JSX?
2. How do you pass data from a parent to a child component in React?
3. Explain the use of the useState hook in React.
4. What is the purpose of the key attribute in list rendering?
5. How do you embed JavaScript expressions in JSX?
6. What does fetch() return in JavaScript?
7. How can you handle events in React?
8. What is the difference between props and state in React?
9. Describe the useEffect hook in React.
10. What is a portal in React?
11. How do you prevent the default action of an event in React?
12. Explain how to render a list of items in React using the .map() function.

13. What does the createPortal function do in React?
14. How do you send a POST request using fetch() in React?
15. Why are props in React read-only?

Long Questions

1. Explain in detail how JSX works in React. Include an example of creating a functional component with JSX.
2. Discuss the useState and useEffect hooks in React. How do they help in managing state and side effects in functional components? Provide examples.
3. How does the fetch() API work in React? Describe how to make GET and POST requests and handle responses, including error handling.
4. What are portals in React? Explain their use cases with an example of implementing a modal using portals.
5. Describe how event handling is managed in React. How does React's synthetic event system work? Provide examples of common event handling techniques.
6. Discuss the differences between props and state in React. How do they interact within a component, and what are the best practices for using them?
7. How can you render lists dynamically in React? Explain the significance of the key attribute and provide an example of rendering a list from an array of objects.
8. What are the benefits and potential pitfalls of using portals in React applications? Provide scenarios where using portals is necessary.
9. Describe the lifecycle of a React component. How do hooks like useState and useEffect play a role in managing the component's lifecycle?
10. Explain how to manage state in both functional and class-based components in React. Compare and contrast the use of useState in functional components with setState in class-based components.

Unit:3:

MCQ:

1. What is the primary purpose of React Router?
    - A) Server-side routing
    - B) Client-side routing
    - C) Data fetching

- D) State management
  Answer: B) Client-side routing

2. Which of the following routers uses the HTML5 history API?
   - A) <HashRouter>
   - B) <MemoryRouter>
   - C) <BrowserRouter>
   - D) <StaticRouter>
     Answer: C) <BrowserRouter>
3. In React Router, which component is used to define the navigation links within an app?
   - A) <Route>
   - B) <Link>
   - C) <Nav>
   - D) <Switch>
     Answer: B) <Link>
4. What is the main purpose of the <Switch> component in React Router?
   - A) To load multiple routes simultaneously
   - B) To display a loading spinner
   - C) To render only the first matching route
   - D) To toggle between different themes
     Answer: C) To render only the first matching route
5. Which syntax is used to define a dynamic route in React Router?
   - A) <Route path="/products" />
   - B) <Route path="/products/:id" />
   - C) <Route path="/products*id" />
   - D) <Route path="/products&:id" />
     Answer: B) <Route path="/products/:id" />
6. Which method is used in Redux to send an action to the store?
   - A) dispatch
   - B) subscribe
   - C) useEffect

- D) connect
  Answer: A) dispatch

7. What does Redux use to manage the state of an application?
   - A) Props
   - B) Reducers
   - C) Context API
   - D) Hooks
     Answer: B) Reducers
8. What is an action in Redux?
   - A) A JavaScript object that describes an event in the app
   - B) A function that directly modifies the state
   - C) A component that renders UI
   - D) A hook for handling side effects
     Answer: A) A JavaScript object that describes an event in the app
9. Which of the following is a core principle of Redux?
   - A) Two-way data flow
   - B) Unidirectional data flow
   - C) Bidirectional data flow
   - D) Multidirectional data flow
     Answer: B) Unidirectional data flow
10. Which method is used to update the state in Redux?
    - A) useState
    - B) dispatch
    - C) setState
    - D) replaceState
      Answer: B) dispatch
11. What does the useSelector hook in Redux do?
    - A) Dispatches actions to the store
    - B) Accesses the Redux state in a component
    - C) Subscribes to Redux store updates

- D) Manages side effects
  Answer: B) Accesses the Redux state in a component

12. What is the role of a reducer in Redux?
    - A) To dispatch actions
    - B) To update the Redux state based on actions
    - C) To handle side effects like network requests
    - D) To manage routing
      Answer: B) To update the Redux state based on actions
13. Which of the following ensures immutability in Redux?
    - A) Modifying the state object directly
    - B) Creating a new state object with changes
    - C) Using local state management
    - D) Bypassing the reducers
      Answer: B) Creating a new state object with changes
14. How are multiple reducers combined in Redux?
    - A) combineReducers
    - B) mergeReducers
    - C) useReducers
    - D) composeReducers
      Answer: A) combineReducers
15. What does the default case in a reducer function do?
    - A) It dispatches a default action
    - B) It returns the current state if no action matches
    - C) It modifies the state directly
    - D) It creates a new state
      Answer: B) It returns the current state if no action matches

---

Short Answer Questions

1. What is the purpose of React Router in a React application?
2. Name two types of routers provided by React Router.

3. How does the <Link> component help in navigation in a React app?
4. What is the use of the <Switch> component in React Router?
5. How do you define dynamic routes in React Router?
6. What is the role of dispatch in Redux?
7. Explain what an action is in Redux.
8. What is the main responsibility of a reducer in Redux?
9. Why is the state in Redux considered immutable?
10. What is the Redux data flow pattern?
11. How does the useSelector hook work in a React component?
12. What does the term "unidirectional data flow" mean in Redux?
13. Explain the role of the default case in a Redux reducer.
14. What is the purpose of error boundaries in React?
15. How can you handle errors using try...catch in React components?

Long Answer Questions

1. Explain the difference between <BrowserRouter> and <HashRouter> in React Router. Provide examples of when you would use each.
2. Discuss the purpose of <Route> in React Router and explain how route parameters work with an example.
3. Describe the use of <Switch> in React Router and how it helps in rendering the first matching route. Provide an example.
4. Explain the concept of nested routes in React Router with an example of how they can be used to create complex page layouts.
5. Discuss the Redux architecture and explain the flow of data from action creation to state update in Redux.
6. What is the role of reducers in Redux, and why must they be pure functions? Provide an example of a simple reducer.
7. Explain how Redux helps in state management in large applications and describe the main benefits of using Redux.
8. Discuss the differences between synchronous and asynchronous error handling in React components, providing examples of each.

9. Explain how combining reducers works in Redux and provide an example of using combineReducers to manage multiple slices of state.
10. Describe how the useSelector and useDispatch hooks work in Redux to interact with the store in a functional React component. Provide examples.

UNIT-4:

MCQS:

1. **What is the primary purpose of a Redux store?**
    - A) To manage local component state
    - B) To handle API requests
    - C) To hold the entire application state in a single place
    - D) To create UI components
      Answer: C) To hold the entire application state in a single place
2. **Which method is used to create a Redux store?**
    - A) createStore()
    - B) initializeStore()
    - C) configureStore()
    - D) buildStore()
      Answer: A) createStore()
3. **What is the purpose of loading initial state in a Redux store?**
    - A) To improve performance
    - B) To populate the store with data before the application runs
    - C) To reset the state
    - D) To manage middleware
      Answer: B) To populate the store with data before the application runs
4. **Which library is commonly used to integrate Redux with React?**
    - A) React-Router
    - B) React-Redux
    - C) Redux-Saga
    - D) Axios
      Answer: B) React-Redux

5. **In Redux, which of the following represents the state of the application?**
    - A) Actions
    - B) Reducers
    - C) Store
    - D) Middleware
      Answer: C) Store
6. **What does the mapStateToProps function do in React-Redux?**
    - A) Maps actions to components
    - B) Maps the Redux state to component props
    - C) Maps component props to state
    - D) Initializes the Redux store
      Answer: B) Maps the Redux state to component props
7. **Which hook can be used to access the Redux store in functional components?**
    - A) useState
    - B) useEffect
    - C) useReducer
    - D) useSelector
      Answer: D) useSelector
8. **What are pure functions in the context of Redux reducers?**
    - A) Functions that always return the same output for the same input
    - B) Functions that perform side effects
    - C) Functions that change external state
    - D) Functions that are asynchronous
      Answer: A) Functions that always return the same output for the same input
9. **What is the primary role of actions in Redux?**
    - A) To define the structure of the state
    - B) To initiate changes in the state
    - C) To render components
    - D) To handle side effects
      Answer: B) To initiate changes in the state

10. **What pattern is commonly used to connect Redux with React components?**
    - A) Higher-Order Components (HOC)
    - B) Functional Programming
    - C) Class Components only
    - D) Context API
      Answer: A) Higher-Order Components (HOC)

Short Questions

1. What is a Redux store, and why is it important?
2. Explain the process of creating and configuring a Redux store.
3. How can you load initial state into a Redux store?
4. Describe how Redux integrates with UI components in React applications.
5. What are the main benefits of using Redux with React?
6. Explain the difference between mapStateToProps and mapDispatchToProps.
7. How do you handle asynchronous actions in Redux?
8. What is the role of reducers in a Redux store?
9. Discuss the React-Redux patterns commonly used in applications.
10. How does the Context API relate to Redux in React?

Long Questions

1. Discuss the architecture of a Redux store, including its key components and their roles.
2. Provide a detailed explanation of how to create and configure a Redux store, including middleware and enhancers.
3. Explain the steps involved in loading initial state into a Redux store, including practical examples.
4. Analyze the process of integrating Redux with React, including the advantages and potential challenges.
5. Describe how to connect Redux to functional components using hooks like useSelector and useDispatch.
6. Compare and contrast Redux with other state management solutions in React, such as Context API and MobX.
7. Discuss the importance of pure functions in reducers and how they contribute to the predictability of application state.

8. Provide examples of common patterns in React-Redux applications and their benefits in managing application state.
9. Explain how to handle complex state transitions in Redux using middleware like Redux Thunk or Redux Saga.
10. Evaluate the impact of using Redux on application performance and scalability, including best practices for optimization.

UNIT-5:

MCQ

1. **What is the primary purpose of middleware in Redux?**
    - A) To manage UI components
    - B) To handle asynchronous actions and side effects
    - C) To define the application state structure
    - D) To optimize performance
      Answer: B) To handle asynchronous actions and side effects
2. **Which of the following is a common use case for Redux middleware?**
    - A) Handling component lifecycle
    - B) Managing state persistence
    - C) Logging actions and state changes
    - D) Rendering UI elements
      Answer: C) Logging actions and state changes
3. **What is the role of logging middleware in Redux?**
    - A) To perform API requests
    - B) To log actions dispatched to the store
    - C) To route components
    - D) To manage application state
      Answer: B) To log actions dispatched to the store
4. **Which middleware would you use for handling asynchronous requests in Redux?**
    - A) Redux Logger
    - B) Redux Thunk
    - C) Redux DevTools

- D) Redux Saga
  Answer: B) Redux Thunk

5. **In Redux, what is a side effect?**
   - A) A state update
   - B) Any operation that interacts with the outside world (e.g., API calls)
   - C) A synchronous action
   - D) A pure function
     Answer: B) Any operation that interacts with the outside world (e.g., API calls)
6. **What is a typical structure for a Redux application?**
   - A) Components, Services, and UI
   - B) Actions, Reducers, and Store
   - C) Routes, Middleware, and APIs
   - D) UI, State, and Logic
     Answer: B) Actions, Reducers, and Store
7. **Which of the following correctly differentiates React, React JS, and React Native?**
   - A) React is a library, React JS is for web, and React Native is for mobile.
   - B) React is a framework, React JS is a library, and React Native is for mobile.
   - C) They are all the same, used for different platforms.
   - D) React is for mobile, React JS is for web, and React Native is for desktop.
     Answer: A) React is a library, React JS is for web, and React Native is for mobile.
8. **Which of the following is an application area of React Native?**
   - A) Building web applications
   - B) Developing mobile applications for iOS and Android
   - C) Creating server-side applications
   - D) Writing desktop applications
     Answer: B) Developing mobile applications for iOS and Android
9. **Which middleware would be best for crash reporting in a Redux application?**
   - A) Redux Logger
   - B) Redux Crashlytics
   - C) Redux Thunk

- D) Redux Promise
  Answer: B) Redux Crashlytics

10. **What is a primary advantage of using middleware in Redux?**
    - A) It enhances the UI
    - B) It separates concerns and promotes code reusability
    - C) It reduces the amount of state in the application
    - D) It simplifies the component structure
      Answer: B) It separates concerns and promotes code reusability

Short Questions

1. What are Redux middleware, and how do they function?
2. Explain the concept of side effects in the context of Redux.
3. How do you create custom middleware in React?
4. List and describe the main types of middleware in Redux.
5. What is the purpose of logging middleware, and how does it work?
6. How does Redux Thunk handle asynchronous actions?
7. Describe the typical structure of a Redux application.
8. What are the differences between React, React JS, and React Native?
9. Identify application areas for React, React JS, and React Native.
10. How can you integrate crash reporting middleware into a Redux application?

Long Questions

1. Discuss the role of middleware in Redux applications and provide examples of how middleware can be used to handle side effects.
2. Explain the process of creating custom middleware in a React application, including code examples.
3. Analyze the different types of middleware available for Redux and their use cases in managing application state.
4. Describe how to handle asynchronous requests in a Redux application, comparing the use of Redux Thunk and Redux Saga.
5. Outline the typical structure of a Redux application, detailing the responsibilities of actions, reducers, and the store.
6. Compare and contrast React, React JS, and React Native in terms of functionality, performance, and typical use cases.

7. Evaluate the application areas of React, React JS, and React Native, discussing the advantages and limitations of each.
8. Discuss the concept of crash reporting middleware in Redux and how it can be implemented in a real-world application.
9. Explain the importance of logging middleware in Redux, including examples of how it can aid in debugging and development.
10. Discuss best practices for organizing a Redux application, including folder structure and separation of concerns.