1. Difference between Class and Functional components, which is more flexible?
2. Class Components are more complex than functional components including constructors, life-cycle methods, render ( ) function and state (data) management. Class components are ES6 classes. Functional components are easy to test. It can have better performance. Functional components are easy to debug. It end up with less code.

**Functional component**

import React from "react";

function FunctionalComponent() {

return <h1>Hello, world</h1>;

}

**Class component**

import React, { Component } from "react";

class ClassComponent extends Component {

render() {

return <h1>Hello, world</h1>;

}

}

1. What is props?
2. Props is the shorthand for Properties in React. They are read-only components which must be kept pure i.e. immutable. They are always passed down from the parent to the child components throughout the application. A child component can never send a prop back to the parent component. This help in maintaining the unidirectional data flow and are generally used to render the dynamically generated data.
3. What is states?
4. React components has a built-in state object.. The state object is where you store property values that belongs to the component.. When the state object changes, the component re-renders.
5. Difference between props and states?

A.

1. How to update the state of the component?
2. State of a component can be updated using this.setState().
3. . What is constructor?
4. The constructor is a method used to initialize an object's state in a class. It automatically called during the creation of an object in a class.

The concept of a constructor is the same in React. The constructor in a React component is called before the component is mounted. When you implement the constructor for a React component, you need to call **super(props)** method before any other statement. If you do not call super(props) method, **this.props** will be undefined in the constructor and can lead to bugs.

7. What is inheritance and how to implement it?

A.React provides the composition and inheritance for code reusability. Inheritance is used to couple components and their properties, which allows an app to implement code sharing between the components.

8. What is axios?

A. React Axios is very easy to modify and is quite lightweight. It also works great with many other frameworks present today. The main purpose of using Axios is to get support for request and response interception, conversion of data into [JSON](https://www.javatpoint.com/json-tutorial) format, and transform it. It also helps you in **protecting XSRF forgery** by default while you request cross-site access.

9. What is react router?

A. React Router is a powerful routing library built on top of React, which helps in adding new screens and flows to the application. This keeps the URL in sync with data that’s being displayed on the web page. It maintains a standardized structure and behavior and is used for developing single page web applications. React Router has a simple API.

10. What's is react hooks?

A. Hooks are named functions that start with the word use and allow us to reuse stateful logic across components without having to write a class component. For example, the useState Hook allows us to add state to a functional component and the useEffect Hook lets us perform side effects like data fetching or updating the document title. Hooks make our code more reusable, easier to understand, and easier to test.”

11. What is useState and useEffect?

A.

12. Difference between react hooks and class?

A.

13. React components lifecycle?

1. ***componentWillMount()***–Executed just before rendering takes place both on the client as well as server-side.
2. ***componentDidMount()***–Executed on the client side only after the first render.
3. ***componentWillReceiveProps()***– Invoked as soon as the props are received from the parent class and before another render is called.
4. ***shouldComponentUpdate()***–Returns true or false value based on certain conditions. If you want your component to update, return **true** else return **false**. By default, it returns false.
5. ***componentWillUpdate()***– Called just before rendering takes place in the DOM.
6. ***componentDidUpdate()***–Called immediately after rendering takes place.
7. ***componentWillUnmount()***– Called after the component is unmounted from the DOM. It is used to clear up the memory spaces.

14. Difference between real DOM and virtual DOM?

|  |  |
| --- | --- |
| **Real DOM** | **Virtual  DOM** |
| 1. It updates slow. | 1. It updates faster. |
| 2. Can directly update HTML. | 2. Can’t directly update HTML. |
| 3. Creates a new DOM if element updates. | 3. Updates the JSX if element updates. |
| 4. DOM manipulation is very expensive. | 4. DOM manipulation is very easy. |
| 5. Too much of memory wastage. | 5. No memory wastage. |

15. What is JSX?

JSX is a shorthand for JavaScript XML. This is a type of file used by React which utilizes the expressiveness of JavaScript along with HTML like template syntax. This makes the HTML file really easy to understand. This file makes applications robust and boosts its performance. Below is an example of JSX:

16.What is arrow function?

17. What is event and how do you create it in react?

A. In React, events are the triggered reactions to specific actions like mouse hover, mouse click, key press, etc. Handling these events are similar to handling events in DOM elements.

cla

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | class Display extends React.Component({      show(evt) {          // code      },      render() {          // Render the div with an onClick prop (value is a function)          return (    <div onClick={this.show}>Click Me!</div>            );      }  }); |

1848 What is import and export?

A.