1. React is primarily used for:
a) Server-side rendering
b) Styling web pages
c) **Building user interfaces**
d) Managing databases

2. JSX allows you to:
a) Write JavaScript inside CSS files
b) Write HTML-like code inside JavaScript
c) Run SQL queries in React
d) Replace Node.js

3. Which of the following is true about props?
a) Props can be modified inside the child component
b) Props allow data flow from child → parent
c) **Props are immutable and flow parent → child**
d) Props are used to manage component state

4. In React, event handlers are written in:
a) snake_case
b) PascalCase
c) camelCase
d) kebab-cas

5. What does React use for efficient DOM updates?
a) Shadow DOM
b) Virtual DOM
c) Real DOM only
d) Web Assembly

6. Write a simple React functional component called HelloWorld that displays "Hello, React!"
:- import React from "react";

```
function HelloWorld() {
  return <h1>Hello, React!</h1>;
}

export default HelloWorld;
```

7. What is the difference between a functional component and a class component?
functional component :- A is a simpler way to define components in React using JavaScript functions. Unlike class components, functional components do functional component not require a constructor or lifecycle methods.
Class components:- Class components are ES6 classes that extend React.Component and can hold and manage internal state.

8. Why should React component names start with a capital letter?

:- JSX will use this capitalization to tell the difference between an HTML tag and a component instance. If the first letter of a name is capitalized, then JSX knows it's a component instance; if not, then it's an HTML element.

9. Explain the difference between JSX and plain JavaScript in React.

:- **JSX**:- A syntax extension that looks like HTML inside JavaScript.
It makes code cleaner and easier to read.

Plain JavaScript: You use functions like React.createElement() to create elements manually.

10.Create a button in React that, when clicked, logs "Button clicked!" to the console.

```
function ClickButton() {

  function handleClick() {

    console.log("Button clicked!");

  }

  return <button onClick={handleClick}>Click Me</button>;

}
```