1.What is the way to change the Functional Component State?

Ans. useState call internally automatically ,useReducer will help you on that.

2.Reducer:

A **reducer** is a function that helps manage state changes in a **structured way** using the useReducer hook.

A **reducer** is a function that:

* Takes **current state** and an **action**.
* Returns **new state** based on the action.

Ex:

const myReducer = (state, action) => {

switch (action.type) {

case "INCREMENT":

return { count: state.count + 1 };

case "DECREMENT":

return { count: state.count - 1 };

default:

return state; // If no action matches, return the same state

}

};

3.Difference between useState and setState?

| **Feature** | **useState (React Hooks)** | **setState (Class Components)** |
| --- | --- | --- |
| **Usage** | Used in **functional components** | Used in **class components** |
| **Syntax** | const [state, setState] = useState(initialValue); | this.setState({ key: newValue }); |
| **State Merging** | Ovveride the state. | **Merges** state automatically |
| **Asynchronous?** | Yes, but no callback function | Yes, supports a callback function |
| **Performance** | Optimized for modern React | Slightly less optimized due to re-renders |