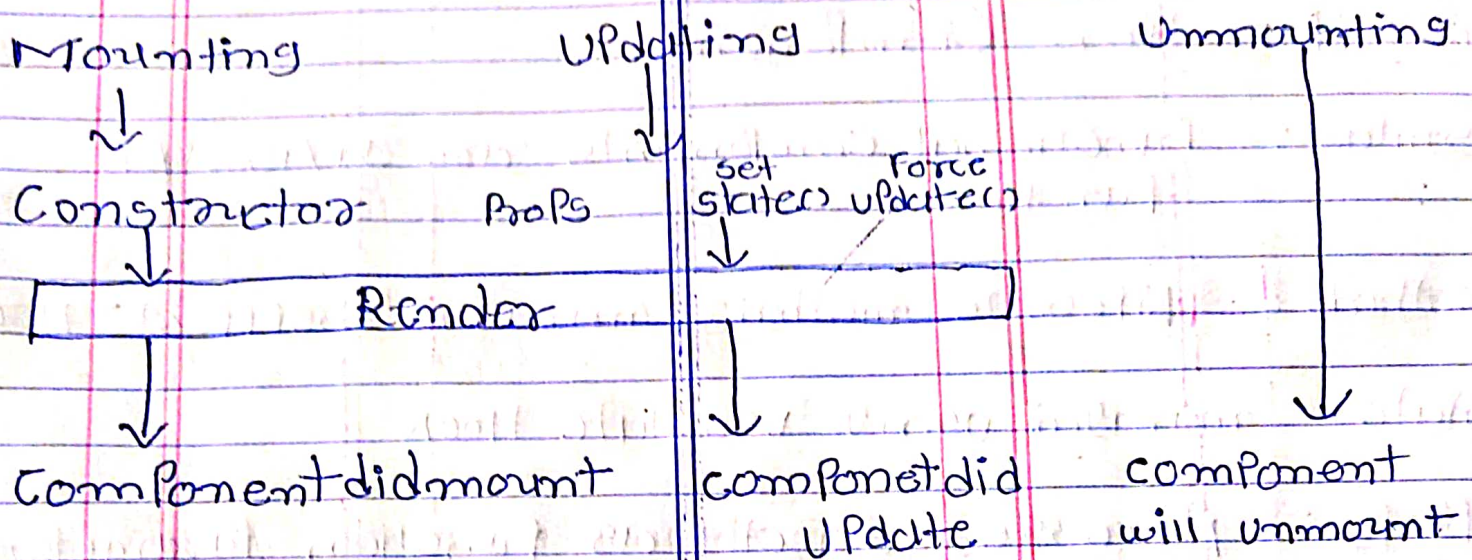


## Module : 4 (List and Hooks)

- React lifecycle is defined as the series of methods that are invoked in different stages of the component's existence. The definition is pretty straightforward but what do we mean by different stages.



### 1) Class Components

- Syntax :- class components are ES6 classes that extend 'React.Component'.
- Methods :- Lifecycle methods are defined as methods within the class.
- state :- Managed using 'this.state' and update 'this.setState()'.
- Props :- Accessed via 'this.props'

Lifecycle methods like 'componentDidMount', 'componentDidUpdate' and 'componentWillUnmount' are used for managing side effects and clean.



- class components can be verbose due to the need to define multiple lifecycle methods and bind methods the constructor.
- Optimization can be more complex requiring lifecycle methods and something manual.

## (2) Function Component

Syntax :- Functional Components are JavaScript Function.

Methods :- Lifecycle methods managed 'useEffect' Hook

State :- managed using useState Hook

Props :- Accessed directly as function parameters.

- Functional Components with Hook 'useEffect' replaces all lifecycle methods can handle. update and unmount logic single interface.

- Functional Components with Hooks Tend to be more concise easier to read with Hook.

- Hooks like 'usememo' and 'usecallback' provide built-in optimizations to unnecessary re-renders and recomputations.

```

1 import React, { Component } from 'react'
2 import { Button } from 'reactstrap'
3
4 export default class Login extends Component {
5   constructor() {
6     super();
7     this.state = {
8       count: 0,
9       id: "",
10      lid: ""
11    };
12  }
13  componentDidMount() {
14
15    let id = setInterval(() => {
16
17      alert("CAN YOU PLEASE LOGIN ");
18      }, 8000);
19    this.setState({ id: id });
20
21    let lid = setInterval(() => {
22      alert("Create a New Account");
23      }, 5000);
24    this.setState({ lid: lid });
25
26  }
27
28  componentDidUpdate(oldprops, oldstate) {
29    if (this.state.id < this.state.lid) {
30      if (this.state.count - oldstate.count === 5) {
31        // alert("CAN DO YOU INCREASE 5POINT");
32        clearInterval(this.state.lid);
33      } else {
34        if (this.state.count - oldstate.count === 1) {
35          // alert("CAN DO YOU INCREASE 5POINT");
36          clearInterval(this.state.id);
37        }
38      }
39    }
40    else {
41      if (this.state.count - oldstate.count === 1) {
42        // alert("CAN DO YOU INCREASE 5POINT");
43        clearInterval(this.state.id);
44      } else {
45        if (this.state.count - oldstate.count === 5) {
46          // alert("CAN DO YOU INCREASE 5POINT");
47          clearInterval(this.state.lid);
48        }
49      }
50    }
51  }
52  render() {
53    return (
54      <div className="d-flex justify-content-center align-items-center ">
55        <div className="text-black">
56          <h1>Count is {this.state.count} </h1>
57          <Button
58            className="me-4 bg-primary border-0"
59            onClick={() => this.setState({ count: this.state.count + 1 })}>
60            >
61            Inc +1
62          </Button>
63          <Button
64            className="bg-primary border-0"
65            onClick={() => this.setState({ count: this.state.count + 5 })}>
66            >
67            Inc +5
68          </Button>
69        </div>
70      </div>
71    );
72  }
73 }
74

```