Question - 1

SCORE: 5 points

Query Based Question 3

React Easy

Consider the following code.

```
import React from "react";

class App extends React.Component {
state = {
name: "",
email: "",
contact: "",
};

setValue = (event) => {};

render() {
const { name, email, contact } = this.state;
return (
<React.Fragment>
        <p>Name = {name}</p>        <p>Email = {email}</p>       {" "}
<p>Contact = {contact}</p>
        <input name="name" onChange={this.setValue} />
        <input name="email" onChange={this.setValue} />
        <input name="contact" onChange={this.setValue} />     {" "}
</React.Fragment>
);
}
}

export default App;
```

There is a common *setValue* function that is used to update all the states on change. Which of the following options provides that functionality?

- (●) setValue = (event) => { this.setState({ [event.target.name]: event.target.value }); };
- ( ) setValue = (event) => { this.setState({event.target.name: event.target.value }); };
- ( ) setValue = (event) => { this.setState({ [event.name]: event.value }); };
- ( ) None of the above

Question - 2

SCORE: 5 points

Class Component Initial Rendering

React Class Components Easy

Consider the following code.

```
import React from "react";

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      Number: 0
    };
  }

  componentDidMount( ) {
    this.increment()
    this.increment()
    this.increment()
  }

  componentWillUnmount(){
    this.increment()
  }

 increment = () => {
    this.setState({
      Number: this.state.Number + 1
    });
  };

  render() {
    return (
      <div>
        <p>Number: {this.state.Number}</p>
        <button onClick={this.increment}>Increment</button>
      </div>
    );
  }
}

export default App;
```

While first rendered, what is displayed in the *p* tag?

- 0
- 1
- 3
- Undefined

Question - 3

SCORE: **5 points**

React Class Component State

React | Class Components | Easy

What is the output?

```
import React from "react"

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = { Number: 0 };
  }
```

```
  componentDidMount() {
    this.setState({ Number: 1 });
    console.log(this.state.Number);
    this.setState({ Number: 2 });
    console.log(this.state.Number);
    this.setState({ Number: 3 }, () => {
      console.log(this.state.Number);
    });
  }

  render() {
    return <div>{this.state.Number}</div>;
  }
}

 export default App;
```

- 0,0,0
- 1,1,1
- 1,2,3
- 0,0,3

Question - 4

Array Functions

SCORE: **5 points**

React | Array Functions | Easy

Consider the following code snippet.

```
import React from "react";

 function App( ) {
              const data = [ { Name: "Ranjit", Roll: 122 },  { Name: "Ron", Roll: 122 } ];
              const display = () => {
                  let result;
                  return (
                              <div>
                                    <h1>{result.Name}</h1>
                                    <h1>{result.Roll}</h1>
                              </div>
                         );
                         };
            return <div>{display()}</div>;
  }

export default App;
```

Which is the preferred method to initialize the result variable to fetch an object named *Ranjit*?

- let result= data.map((data) => data.name=="Ranjit");
- let result= data.filter((data) => data.name=="Ranjit");
- let result= data.find((data) => data.name=="Ranjit");
- None of the above

Question - 5

SCORE: **5 points**

State Update

React | React Hooks | Functional Components | Easy

Consider the following code.

```
import React, { useState } from "react";

const App = () => {
  const [student, setStudent] = useState({
    Name: "Abc",
    Roll: "123",
    Presence: 0,
  });

  const update = (Presence) => {
  };

  return (
    <>
      <p>{student.Presence}</p>
      <button onClick={() => update(student.Presence + 1)}>
        Update Student
      </button>
    </>
  );
};

export default App;
```

Which option increments the student presence by 1?

- this.setState({Presence});
- setStudent(pre=>{...pre,Presence});
- setStudent({ ...student, Presence });
- None of the above

Question - 6

SCORE: **5 points**

Initial Rendering

React | Easy | Functional Components | State Management

What is the output of the following code?

```
import React, { useState } from "react";

const App = () => {
  const [number, setNumber] = useState(0);
  console.log(number);//output
  return (
    <>
      <p>{number}</p>
      <button onClick={setNumber((pre) => pre + 1)}>increment</button>
```

```
    </>
  );
};

export default App;
```

- 0
- 1
- undefined
- Error

Question - 7

Event

SCORE: **5 points**

Event Handlers | Easy

Which React code snippet returns the mouse position?

-
```
useEffect (() => {
        const setFromEvent = (e) => setPosition({ x:
e.clientX, y: e.clientY });
window.addEventListener("mousemove",
setFromEvent);

        return () => {
                window.removeEventListener("mousemove",
setFromEvent);
        };
}, []);
```

-
```
useEffect (() => {
        const setFromEvent = (e) => setPosition({ x:
e.clientX, y: e.clientY });
window.addEventListener("mousemove",
setFromEvent);
}, []);
```

- Both of the snippets above will provide the functionality required.
-
```
useCallback(() => {
        const setFromEvent = (e) => setPosition({ x:
e.clientX, y: e.clientY });
        window.addEventListener("mousemove",
setFromEvent);

        return () => {
                window.removeEventListener("mousemove",
setFromEvent);
};
}, [])
```

- None of the above

Question - 8

SCORE: **5 points**

State Management

State Management | Easy

Which change to the following code will cause it to increment *count* every second.

```
function Counter() {
 let [count, setCount] = useState(0);
 useEffect (() => {
  let id=setInterval(() => {
   setCount(count + 1);
 }, 1000);
return () => clearInterval(id);
}, []);
return <h1>{count}</h1>;
}
```

- Replace line 5 with setCount(++count);
- Replace line 5 with setCount(prevCount+ 1);
- Replace line 5 with setCount(count++);
- Replace line 5 with setCount(prevCount => prevCount+ 1);
- None of the above

Question - 9

SCORE: **5 points**

Components

Easy | Class Components

This code is part of a React application.

```
const Home = () => {
  const [isLoading, setLoading] = useState(true);
  if(isLoading) {
     return <Loader />;
  }
  useEffect (() => { //fetchData }, []);
}
```

Which of the following statements is correct?

- The isLoading state has been initialized to TRUE, but it should be FALSE initially.
- An error will be thrown as the useEffect hook is called conditionally.
- The Loader component will always be shown, and the useEffect hook will not be executed.
- None of the above

Question - 10

SCORE: **5 points**

React: True statements

React | Front-End Development | Easy

Which of the following statement(s) is true about ReactJS?

1. Only one root element can be returned per component.
2. Both arguments of the useEffect hook are compulsory.
3. The useState hook returns an object.
4. The useState hook cannot be used outside of a functional component.

- [x] 1
- [ ] 2
- [ ] 3
- [x] 4

Question - 11

SCORE: **5 points**

React: Route

React | Front-End Development | Router | Easy

```
import React from 'react';
import {BrowserRouter as Router, Switch, Route} from "react-router-dom";
import Home from "./Home";
import About from "./About";

function App() {
  return (
    <Router>
      <Switch>
        <Route path="/" component={Home} />
        <Route path="/about" exact component={About} />
      </Switch>
    </Router>
  )
}

export default App
```

What component will get rendered if the URL is "http://localhost:3000/about"?

- [x] <Home/ >
- [ ] <About />
- [ ] <Home /> <About />
- [ ] No output

Question - 12

SCORE: **5 points**

React: Children

React | Front-End Development | Props | Easy

What is the output?

```
import React from 'react'

function Test (props) {
  return (
    <div>
      <p>Hello {props.name}</p>
      <p>{props.children}</p>
    </div>
  );
}
export default function App() {
  return (
    <Test name="candidate" children="Frontend">
      <p>Welcome to Hackerrank!</p>
    </Test>
  )
}
```

- Hello candidate
  Frontend
- Hello candidate
  Welcome to Hackerrank!
- Hello candidate
  undefined
- Error

Question - 13

SCORE: **5 points**

React: Check statements

React | Front-End Development | Props | Easy

```
import React from 'react';
function Sample(props) {
  return (
   <>
    <div>
        <p>{props.name}</p> {/* Statement 1 */}
        <p>{props[designation]}</p> {/* Statement 2 */}
    </div>

     <div>
        <p>{props[0]}</p> {/* Statement 3 */}
        <p>{props['designation']}</p> {/* Statement 4 */}
    </div>
  </>
  )
```

```
}
export default Sample;
```

Which of the statement(s) will not generate an error? ('name' and 'designation' are the props passed to the <Sample /> from the App component)

- [x] Statement 1
- [ ] Statement 2
- [x] Statement 3
- [x] Statements 4

Question - 14

SCORE: **5 points**

React: Welcome

React | Front-End Development | Easy

```
import React from "react";
function App() {
  return (
   <div>
    <p>Hi</p>
    <p>Welcome to Hackerrank</p>
   </div>
   <div>
    <p>This is a react problem!</p>
   </div>
  );
}
export default App;
```

What is the output of the code snippet?

- [ ] Hi
  Welcome to Hackerrank
- [ ] Hi
  Welcome to Hackerrank
  This is a react problem!
- [ ] This is a react problem!
- [x] This code will generate an error.

Question - 15

SCORE: **5 points**

React: Add class

Front-End Development | React | Classes | Easy

Which of the following code snippets can be used to dynamically add a class to a tag in ReactJS? Assume all imports and exports are there.

S1:

```
function App() {
  const [state, setState] = useState(false);
```

```
  return (
    <div>
    <button onClick={() => setState(!state)}>Click me!</button>
    <h1 className={state && "red-color"}>Hello</h1>
    </div>
  );
}
```

S2:

```
function App() {
  const [state, setState] = useState(false);
  const color = "red-color";
  return (
    <div>
    <button onClick={() => setState(!state)}>Change color</button>
    <h1 className={color}>Hello</h1>
    </div>
  );
}
```

S3:

```
function App() {
  const [state, setState] = useState(false);
  return (
    <div>
    <button onClick={() => setState(!state)}>Click me!</button>
    <h1 className={state ? "red-color": "blue-color"}>Hello</h1>
    </div>
  );
}
```

S4:

```
function App() {
  const [state, setState] = useState(false);
  return (
    <div>
    <button onClick={() => setState(!state)}>Click me!</button>
    <h1 className="red-color">Hello</h1>
    </div>
  );
}
```

- (●) S1
- ( ) S2
- (●) S3
- ( ) S4

Question - 16

SCORE: **5 points**

React: Hooks and Their Applications

React | Hooks | Easy

Match the commonly used React hooks and their applications.

Hook	Application
1. useState	a. Perform side effects on updates
2. useEffect	b. Returns memorized values
3. useContext	c. Used to track state
4. useMemo	d. Manage state globally

- 1.b
  2.c
  3.d
  4.a

- 1.c
  2.a
  3.d
  4.b

- 1.d
  2.c
  3.a
  4.b

- 1.b
  2.d
  3.a
  4.c

Question - 17

SCORE: **5 points**

React: Styling in JavaScript

React CSS3 Easy Styling React

Select the appropriate option to generate inline styles equivalent to the CSS stylesheet.

```
.my-style{
    background-color: blue;
    color: red;
    font-size: large;
    border: solid 10px black;
}
```

```
const MyComponent = () => {
        return (
                <div style="background-color: blue;
                                color:red;
                                font-size:large;
                                border:solid 10px black; ">
                        My CSS Div
                </div>
                )
};
```

```
const MyComponent = () => {
        return (
                <div style={background-color: "blue",
                                color:"red",
                                font-size:"large",
                                border:"solid 10px black"}>
                        My CSS Div
                </div>
        )
};
```

```
const MyComponent = () => {
        return (
                <div style={{background-color: "blue",
                                color:"red",
                                fontSize:"large",
                                border: "solid 10px black"}}>
                        My CSS Div
                </div>
        )
};
```

```
const MyComponent = () => {
        return (
                <div style={'background-color': "blue",
                                'color':"red",
                                'fontSize':"large",
                                'border': "solid 10px black"}>
                        My CSS Div
                </div>
        )
};
```

Question - 18

SCORE: **5 points**

React: Valid Routes

React | React Router | Routes | Easy

Which route is not valid for this React Router component?

```
<Routes>
        <Route path='/' element={<Home />} />
        <Route path='/user/:id' element={<User />} />
        <Route path='/user/:id/edit' element={<UserEdit />} />
        <Route path='/post/:id' element={<Post />} />
        <Route path='/posts' element={<Posts />} />
</Routes>
```

- /user/1/edit
- /post/10

- [x] /post
- [ ] /profile/12

Question - 19

SCORE: **5 points**

React: Components and Routes

React | Easy | React Router | Routes

Which component will load when the Route '/signin' is requested from this React Router?

```
<BrowserRouter>
    <Routes>
        <Route path="/" element={<LandingPage />} />
        <Route path="/home" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
        <Route path="/contact" element={<ContactPage />} />
        <Route path="/login" element={<LoginPage />} />
        <Route path="/signup" element={<SignupPage />} />
        <Route path="*" element={<NotFoundPage />} />
    </Routes>
</BrowserRouter>
```

- [ ] LoginPage
- [ ] SignupPage
- [x] NotFoundPage
- [ ] HomePage

Question - 20

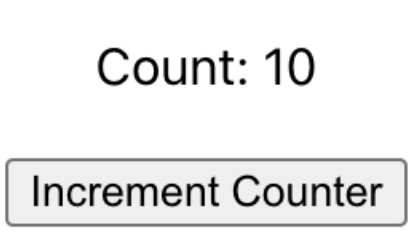
SCORE: **5 points**

React: Counter

React | Hooks | Easy

```
const Counter = () => {
        const [number, setNumber] = React.useState(0);
        const incrementNumber = () => {
                setNumber(number + 1);
        };
        React.useEffect(() => setNumber(10), [number]);
        return (
                <div>
                        Count: {number}
                        <button onClick={() => incrementNumber()}>Increment Counter</button>
                </div>
        )
};
export default Counter;
```

The ReactJS Counter component is rendered as shown. The value of *{numberr}* is 10. What will be the value of *{number}* after clicking "Increment Counter" 3 times?



- 15
- 13
- 10
- 0

Question - 21

SCORE: **5 points**

React: Functional Components

React Easy Syntax

Which is the functional component that is equivalent to this React Class component?

```
class Hello extends React.Component{
    render(){
        const {name, age} = this.props;
        return(
            <div>
                 Hello {name} is {age} years old.
            </div>
        )
    }
}

Hello.propTypes = {
    name: PropTyps.string,
    age: PropTypes.number
}

Hello.defaultProps = {
    name: 'David'
}

export default Hello;
```

-

```
function Hello({ name = 'David', age }) {
        return (
                        <div>
                                <p>Hello {name} is {age} years old</p>
                        </div>
                );
}
Hello.propTypes = {
        name: PropTypes.string,
        age: PropTypes.number.isRequired
```

```
};
export default Hello;
```

```
function Hello({ name, age }) {
        return (
                        <div>
                                <p>Hello {name} is {age} years old</p>
                        </div>
                );
}
Hello.propTypes = {
        name: PropTypes.string,
        age: PropTypes.number.isRequired
};
export default Hello;
```

```
function Hello({ name = 'David', age }) {
        return (
                        <div>
                                <p>Hello {name} is {age} years old</p>
                        </div>
                )
}
Hello.propTypes = {
        name: PropTypes.string,
        age: PropTypes.number.isRequired
};
export default Hello;
```

The Class component cannot be converted into a functional component.

Question - 22

SCORE: **5 points**

React: States and Props

React | Component State Management | State Management | Easy

Consider these React components as rendered below.

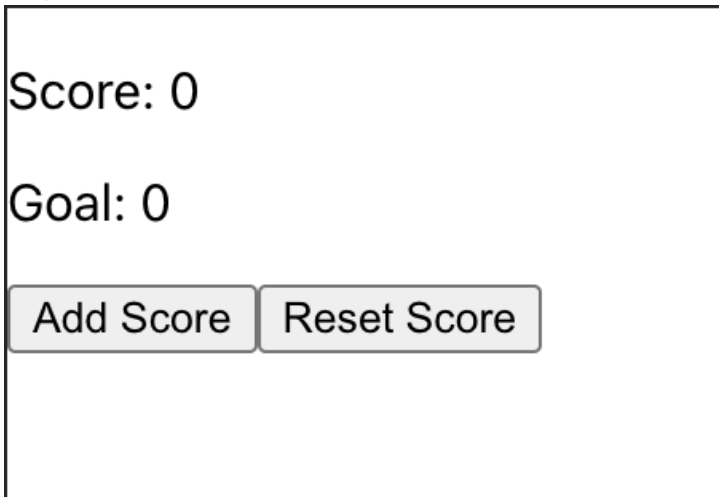
ScoreBoard.js

```
const ScoreBoard = () => {
        const [score, setScore] = React.useState(0);
        const [goal, setGoal] = React.useState(0);

        const newScore = (event) => {
                setScore(score + 1);
                setGoal(goal + 1);
        }
        return (
                <div>
                        <p>Score: {score}</p>
                        <p>Goal: {goal}</p>
                        <ScoreManager newScore={newScore} score={score} goal={goal} />
                </div>
        )
}

export default ScoreBoard;
```

ScoreManager.js

```
const ScoreManager = (props, { score, goal}) => {
        const resetScore = (event) => {
                score = 0
                goal = 0
        }
        return (
                <div>
                        <button onClick={props.newScore}> Add Score </button>
                        <button onClick={resetScore}> Reset Score </button>
                </div>
        )
};
export default ScoreManager;
```

Output:



What will be the values of Score and Goal printed after clicking the 'Add Score' button 5 times and then clicking the 'Reset Score' button

- Score: 5
  Goal: 5
- Score: 0
  Goal: 0
- Score: 4
  Goal: 4
- Score: 5
  Goal: 0

Question - 23

SCORE: **5 points**

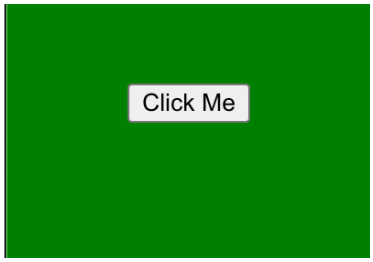
React: Events

React EventListener Easy

```
import React from 'react';
import PropTypes from 'prop-types';
import './Question2.css';

const GoalComponent = () => {
        const shoot = (message, event) => console.log(message);
        return (
```

```
                <div className="Question2">
                        <div className='main-div' onClick={(event) => shoot("No Goal", event)}>
                                <button onClick={(event) => shoot("Goal", event)} >
                                        Click Me
                                </button>
                        </div>
                </div>
        )
}

export default GoalComponent;
```

The given React component renders as shown below.



What is logged to the console when "Click Me" is clicked?

- Goal
- No Goal
- Goal
  No Goal
- No Goal
  Goal

Question - 24

SCORE: **5 points**

Render Function

Easy | React | React Component Lifecycle

Which of the following functions gets invoked after the render() function when `state` or `props` is updated?

- componentWillMount()
- componentDidMount()
- componentWillReceiveProps()
- componentDidUpdate()

Question - 25

SCORE: **5 points**

Method Search

Easy | React | React Component Lifecycle

Which of the following methods is/are invoked when there is an error during rendering in the React lifecycle?

- [ ] getDerivedStateFromError()
- [ ] componentDidCatch()
- [x] Both A and B.
- [ ] None of the above

Question - 26

SCORE: **5 points**

Wrap Code

Easy React

Assuming options state values are given in the constructor, which code should wrap the components that are going to have access to the context inside a class component?

- [x] <NotesContext.Provider value={this.state.notes}> //code </NotesContext.Provider>
- [ ] <NotesContext value={this.state.notes}> //code </Notes.Provider>
- [ ] <NotesContext.Provider value={this.state.notes}> //code <Notes.Provider/>
- [ ] <NotesContext.Provider value={this.state.notes}/> //code <Notes.Provider>

Question - 27

SCORE: **5 points**

Submit

Easy React

What element type does this render?

```
function Music() {
  return <span>Tiesto - Surrounded by Light</span>;
}

ReactDOM.render(<Music/>, document.getElementById('root'));
```

- [ ] div
- [x] span
- [ ] Component
- [ ] None above

Question - 28

SCORE: **5 points**

Hooks

Easy React

Which is the equivalent code using hooks?

```
import React, { Component } from 'react'
export default class SampleComponent extends Component {
  componentDidMount() {
   //
  }
  render() {
   return (<div>Hello</div>)
  }
}
```

- import React, { useEffect } from 'react'
  const SampleComponent = () => {
   useEffect(() => {
    //
   }, [this]);
   return ('Hello')
  }
  export Component
- import React, { useEffect } from 'react';
  Default const SampleComponent= () => { Effect(() => { {/.../} }, []); return ('Hello') } export SampleComponent
- import React, { useEffect } from 'react';
  const SampleComponent = () => {
    useEffect(() => { // }, []); return ('Hello');
  } export SampleComponent
- None above

Question - 29

SCORE: **5 points**

Click 3

Easy React

What is displayed in H1 when after a user clicks the button 3 times?

```
import React, { useState } from 'react';
export default function App() {
  const [clicked, setClicked] = useState(3);
  return (
    <div className="App">
      <button onClick={() => setClicked((clicked - 1) * 2)}>
        Click Me
      </button>
      <h1>{clicked}</h1>
    </div>
  );
}
```

- 5
- 6
- 14
- 10

Question - 30

SCORE: **5 points**

Science Project

Easy React

Which of the following is a valid syntax to use the variable inside a component?

```
const SkyColor = "Blue";
```

- <Uppersky className="Space_Details">${Skycolor}</Uppersky>
- <Uppersky className="Space_Details"><Skycolor></Skycolor> </Uppersky>
- <Uppersky className="Space_Details" class="SkyColor"></Uppersky>
- <Uppersky className="Space_Details">{SkyColor}</Uppersky>

Question - 31

SCORE: **5 points**

Hello

React React Component Easy

What is the output generated by the App component?

App.js

```
import Component from "./Component";
function App() {
  return(
    <>
      <Component title = "Hello"/>
      <Component title = "Everyone"/>
      <Component title = "Good Morning"/>
    </>
  );
}
export default App;
```

Component.js

```
export default function Component(props){
  return(
    <div>
      <h1>{props.title}</h1>
    </div>
```

```
  );
}
```

- Hello
  Hello
  Hello
- Hello
  Everyone
  Good Morning
- This code wouldn't compile
- Hello
  Exception thrown

Question - 32

SCORE: **5 points**

React State

React Easy

Which option replaces "//insert code" to update the state of a component multiple times with `newState` function calls?

```
export default class Game extends React.Component{
  constructor(props){
    super(props);
    this.state = { name : "Peter" };
    this.newState = this.newState.bind(this)
  }
  newState(){
    //insert code
  }

}
```

- this.setState="David";
- this.setChangeState({name:"David"})
- this.setState({name:"David"})
- Both A and B

Question - 33

SCORE: **5 points**

Name

React React Component Easy

Assuming this component is imported and used on a page, what is its output?

```
import React, { Component } from 'react';
class App extends Component {
```

```
  constructor(){
   super();
   this.state={name:"peter"}
  }
  render(){
    return (
      <div>
        {this.state.name}
      </div>
    )
  }
} export default App
```

- This code will not compile.
- Error
- Peter
- peter

Question - 34

SCORE: **5 points**

Clock

Easy | React | React Component Lifecycle

This component is used in a page where a div tag with root ID exists. What is the output of the following code?

```
import React from 'react';
import ReactDOM from 'react-dom';
export default class Clock extends React.Component {
  constructor(props){
    super(props);
    this.state = { time : new Date() };
  }
  render(){
    return (
      <div><h1>{ this.props.title } !</h1>
      <h2>{this.state.time}</h2></div>);
  }
}
ReactDOM.render(
  <Clock title="ThisisMyClock" />,
  document.getElementById('root')
);
```

- {CurrentTimeinFormat} ThisisMyClock
- This code will not compile.
- ThisisMyClock {CurrentTimeinFormat}
- None of above

Question - 35

SCORE: **5 points**

Leap Year

Easy React

The parameter *year* is passed as a prop. What code replaces "//CODE HERE" to complete the function?

```
let LeapYear=(props)=>{
 //CODE HERE
 if((year % 4 === 0 && year % 100 !== 0) || year % 400 === 0){
   return <p>{year} is a leap year.</p>
 }else{
   return <p>{year} is not a leap year.</p>
 }
}
export default LeapYear
```

- let year;
- let year = props.year;
- year = props.year;
- let props.year;

Question - 36

SCORE: **5 points**

Attribute Access

Easy React

Which option will access attributes of the CSS file at "./mystyle.module.css"?

```
.TextDesign {
   colour: Blue;
   padding: 40px;
   font-family: Arial;
   text-align: centre;
}
```

- import React from 'react';
  import ReactDOM from 'react-dom';
  import styles from './mystyle.module.css';
  class TextColor extends React.Component {
    render() {
      return <h1 className=styles>Hello I am a text!;</h1> }
  }
  export default TextColor;

- import React from 'react';
  import ReactDOM from 'react-dom';
  import styles from './mystyle.module.css';
  class TextColor extends React.Component {

```
 render() {
  return <h1 id={styles.color}>Hello I am a text!</h1>;
 }
}
export default TextColor;
```

```
import React from 'react';
import ReactDOM from 'react-dom';
import styles from './mystyle.module.css';
class TextColor extends React.Component {
 render() {
  return <h1 className={Textdesign.color}>Hello I am a text!</h1>;
 }
}
export default TextColor;
```

```
import React from 'react';
import ReactDOM from 'react-dom';
import styles from './mystyle.module.css';
class TextColor extends React.Component {
 render() {
  return <h1 className={styles.TextDesign}>Hello I am a text!</h1>;
 }
}
export default TextColor;
```