Question - 1

React Hooks

SCORE: **5 points**

React | React Hooks | Medium

Which *useEffect* runs infinitely when the page is loaded?

A) useEffect(() => { console.log("updated 1"); });
B) useEffect(() => { console.log("updated 2"); },[]);

○ A

○ B

○ Both A and B

● None of the above

Question - 2

Query Based Question 2

SCORE: **5 points**

React | Medium | Class Components

```
import { object } from "prop-types"
import React from "react";

class App extends React.Component {
  state = {
    count: 0,
  };
  add = () => {
    const { count } = this.state;
    this.setState({ count: count + 1 });
    console.log("Count is incremented", this.state.count);
  };

  render() {
    return (
      <React.Fragment>
        <button onClick={this.add}>Add</button>
      </React.Fragment>
    );
  }
}

export default App;
```

Which of the options will print the updated count state while calling the *add* function?

○ add() { const { count } = this.state; await this.setState({ count: count + 1 }); console.log("Count is incremented", this.state.count); }

○ async add() { const { count } = this.state; this.setState({ count: count + 1 }); console.log("Count is incremented", this.state.count); }

● add() { const { count } = this.state; this.setState({ count: count + 1 }, () =>{ console.log("Count is incremented", this.state.count); }); }

○ None of the above

Question - 3

SCORE: **5 points**

Query Based Question

React | React Hooks | Medium

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

const App = () => {
  const [number, setNumber] = useState(0);

  const incrementNumber = () => {
    setNumber(number++);
  };

  return (
    <div>
        <p>Number: {number}</p>
            <button onClick={incrementNumber}>Increment Number</button>

    </div>
  );
};

export default App;
```

Which of the following is true about the code snippet?

○ The incrementNumber function increments the number by one.

○ The incrementNumber function decrements the number by one.

○ The incrementNumber function updates the value of the number to undefined.

● The incrementNumber function throws an error.

Question - 4

SCORE: **5 points**

React Component

React | React Hooks | Medium

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

const app = () => {
  const [number, setNumber] = useState(12);
  return (
    <div>
      <p>Number:- {number}</p>
      <button onClick={() => setNumber(number + 1)}>+</button>
    </div>
  );
};

export default app;
```

Which of the following statements is true about this code snippet?

○ The setNumber function is called incorrectly.

● A React functional component should have a capitalized name.

○ The code is correct.

○ setNumber is not declared correctly.

Question - 5

HOC

SCORE: **5 points**

HOC | React | Medium

Which value of the parameter *test* will cause the message *hello* to be displayed?

```
const withHoC = (C, test = 'hello’) =>
 test ? ({ message }) => <C message={message + test} /> : C;

const HocedC = withHOC(
 ({ message = ‘hello’ }) => <div>{message}</div>,
 /* value of test here */
);

const MyApp = () => <HocedC message='hello’ />;
```

○ null

● 'hellohello'

○ undefined

○ any string value

Question - 6

Reducer

SCORE: **5 points**

Reducer | React | Medium

What will be the value of the *counter* after the button is clicked?

```
const reducer = (state, action) => {
 switch (action. type) {
   case ‘increment’: {
     return state + 1;
   }
   case ‘double’: {
     return state * 2;
   }
   default: {
     return state;
   }
 }
};
const MyComponent = () => {
 const [counter, dispatch] = useReducer(reducer,0);
 const fire = () = {
   ['increment', ‘double’, ‘reset’, “increment '].forEach((type) => dispatch({ type }));
 };
 return <button onClick={fire}>{counter}</button>;
};
```

○ 0

○ 1

● 3

○ undefined

Question - 7

SCORE: **5 points**

React - Components 2

Functional Components | Medium

A *<BankDashboard/>* component includes a function *calculateEmi()* which does expensive calculations using financial data. The function performs its heavy calculation on every re-render which happens due to state changes inside the component. It causes the dashboard to load a bit slowly.

What change should be made to optimize the component's performance?

○ Use StrictMode as it activates additional checks and warnings for its descendants.

○ Wrap the component with the React.memo higher-order component.

○ Wrap the calculateEmi() function with the useContext hook.

● Wrap the calculateEmi() function with the useMemo hook.

○ Implement lazy loading for the component.

Question - 8

SCORE: **5 points**

Hooks 2

React Hooks | Medium

A developer is struggling to resolve a warning with the following function in the repository.

```
function List (props) {

  const [counter, setCounter] = useState(0);


  useEffect (() => {

    setCounter(counter)

   }, []);

return <div>Counter: {counter} </div>;
}
```

Which of the following is true about the warning?

○ The warning can be resolved by adding the counter in the dependency array which is passed as a second argument to the useEffect().

○ The warning can be resolved by removing the dependency array which is passed as a second argument to the useEffect() function.

○ The warning can be suppressed by adding // eslint-disable-line react-hooks/exhaustive-deps after the useEffect().

○ The warning is about useEffect(). There is a counter inside the useEffect() function with an empty dependency array as a second argument.

● All of the above

Question - 9

Hooks

SCORE: 5 points

Hooks | React Hooks | Medium

```
function App() {
  const counterFn = () => {
    const [counter, setCounter] = useState(0);
    setCounter(counter++)
  }

return (
<>
  <div>
    <button onClick={counterFn}>Incr Counter</button>
  </div>
 </>
 )
}
```

What is the problem with this code?

○ The setCounter() function cannot be passed a variable with an operator. A static value should be passed instead.

●

This will result in the error "React hooks must be called in a React function component or a custom React hook function".

○ The code has a compile time exception and React cannot build the project.

○ This causes an unhandled exception.

Question - 10

SCORE: **5 points**

React Props

Data Passing | React Props | Medium

How can the component *MyChild1* access the *test* variable without using props?

```
const MyChild1 = () => {
 return <div>{test}</div>;
};

const MyParent = () => {
 const test = 'my value’;
 return <MyChild1 />;
};
```

○ It is not possible with React; we need to use props.

●

```
const Context = React.createContext();
const MyChild1 = () => {

        const test = useContext(Context);
        return <div>{test}</div>;
};

const MyParent = ()
 const test = 'my value'
 return (
        <Context.Provider value={test}>
                <Mychild1 />
        </Context.Provider>
 );
};
```

○

```
const MyParent = () => {
 const test = ‘my value';
 return (
        <React.Context value={test}>
                <MyChild1 />
        </React.Context>
 );
};
```

○

```
const MyChild1 = (props, {test}) => {
        return <div>{test}</div>;
};
```

```
const MyParent = () => {
 const test = 'my value';
 React .createContext(test)
 return ( <MyChild1 /> );
};
```

Question - 11

Infinity

SCORE: **5 points**

Medium React

Which of the following options will display values from 1 to 10 ([1,10)) infinitely?

●

```
import React, { useState, useEffect } from 'react';
export default function App() {
 const [a, change_a] = useState(1);
 useEffect(() => {
  setTimeout(() => {
   change_a((a) => a + 1); },
  1000);
  if(a>10){
   change_a((a) => 1);
  }
 } );
 return ({a});
}
```

○

```
import React, { useState, useEffect } from 'react';
export default function App() {
 const [a, change_a] = useState(1);
  useEffect(() => {
   setTimeout(() => { change_a((a) => a + 1); }, 1000);
  if(a>11){
   change_a((a) => 1);
  }
 });
 return ({a});
}
```

○

```
import React, { useState, useEffect } from 'react';
export default function App() {
 const [a, change_a] = useState(1);
 useState(() => {
  setTimeout(() => { change_a((a) => a + 1); },
  1000);
  if(a>10){
   change_a((a) => 1);
  }
 });
 return ({a});
}
```

○ None above

Question - 12

Snippet

SCORE: **5 points**

Medium React React Component Lifecycle

What is the output of this component?

```
export default class Clock extends React.Component {
 constructor(){
  this.setState({time: new Date()});
 }
  componentDidMount() {
    this.setState({ time : "Time never changes"});
  }
 render() {
   return <h1>Look at the time: {this.state.time}</h1>;
 }
}
```

○ {CurrentDateTime}

○ Time never changes

● This code will not compile

○ None of above

Question - 13

Data

SCORE: **5 points**

Medium React

The component "Data.js" is imported and used with "App.js" and used on a page. What is the output of the "App" component?

App.js

```
import Data from "./Data";
function App() {
 return (
<>
<h1> The data is shown below </h1>
 <Data/>
 </>
 );
}

export default App;
```

Data.js

```
export default function Data (props) {
 return (
   <div>
     <h1>I am an H1 title</h1>
     <p> I am a paragraph. </p>
     <h1>{props.title}</h1>
   </div>
 );
}
```

```
Data.defaultProps={
 title:"I am the default"
};
```

○ I am an H1 title
I am a paragraph.

●
I am an H1 title
I am a paragraph.
I am the default

○
I am a paragraph
I am the default

○ This code will not compile.

Question - 14

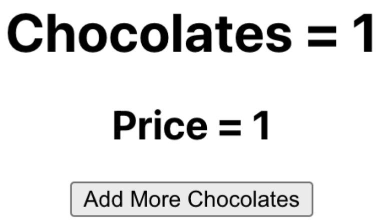
SCORE: **5 points**

React: Hooks I

React | Front-End Development | Medium | JavaScript

```
import { useEffect, useState } from "react";
export default function App() {
  const [noOfChocolates, setChocolates] = useState(1);
  const [chocolatePrice, setChocolatePrice] = useState(1);
  const addChocolates = function(){
    setChocolatePrice(num2=>num2+1)
  }
  useEffect(() => {
    setChocolates(1);
    setChocolatePrice(1);
  });
  return (
    <div className="my-app">
      <h1>Chocolates = {noOfChocolates}</h1>
      <h2>Price = {chocolatePrice}</h2>
      <button onClick={addChocolates}>Add More Chocolates</button>
    </div>
  );
}
```

The react component for a chocolate shop renders as shown below.



Chocolates = 1

Price = 1

What are the values of *Chocolates* and *Price* after the "*Add More Chocolates*" button is clicked 5 times?

● Chocolates = 1
Price = 1

○ Chocolates = 6
Price = 6

○ Chocolates = 1
Price = 6

○ Chocolates = 6
Price = 1

Question - 15

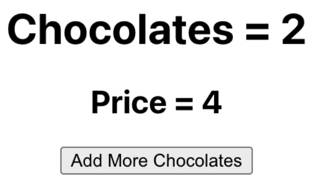
React: Hooks

SCORE: **5 points**

React | Front-End Development | Medium | Front-End Frameworks | JavaScript

```
import { useEffect, useState } from "react";
export default function App() {
  const [noOfChocolates, setChocolates] = useState(1);
  const [chocolatePrice, setChocolatePrice] = useState(1);
  const addChocolates = function(){
    setChocolatePrice(num2=>num2+1)
  }
  useEffect(() => {
    setChocolates(1);
    setChocolatePrice(1);
  });
  return (
    <div className="my-app">
      <h1>Chocolates = {noOfChocolates}</h1>
      <h2>Price = {chocolatePrice}</h2>
      <button onClick={addChocolates}>Add More Chocolates</button>
    </div>
  );
}
```

The react component for a chocolate shop renders as shown below.



Chocolates = 2

Price = 4

What are the values of *Chocolates* and *Price* after the "*Add More Chocolates*" button is clicked 5 times?

○ Chocolates = 7
Price = 4

○ Chocolates = 7
Price = 9

● Chocolates = 2
Price = 9

○ Chocolates = 2
Price = 4