

**Started on** Tuesday, 27 May 2025, 2:39 PM**State** Finished**Completed on** Tuesday, 27 May 2025, 2:46 PM**Time taken** 7 mins 35 secs**Marks** 5.00/5.00**Grade** 100.00 out of 100.00**Question 1**

Complete

Mark 1.00 out of 1.00

What is logged repeatedly in this example?

```
function useLogger(value) {  
  React.useEffect(() => {  
    const id = setInterval(() => {  
      console.log("Value is:", value);  
    }, 1000);  
    return () => clearInterval(id);  
  }, []);  
}
```

```
function App() {  
  const [count, setCount] = React.useState(0);  
  useLogger(count);  
  return <button onClick={() => setCount(count + 1)}></button>;  
}
```

- ☐ a. A runtime error
- ☒ b. Always 0
- ☐ c. Updated count value each second
- ☐ d. Undefined

**Question 2**

Complete

Mark 1.00 out of 1.00

What is printed every second after clicking the button a few times?

```
function App() {  
  const [count, setCount] = React.useState(0);  
  const log = () => {  
    console.log("Count is:", count);  
  };  
  React.useEffect(() => {  
    const id = setInterval(log, 1000);  
    return () => clearInterval(id);  
  }, []);  
}
```

```
  return <button onClick={() => setCount(count + 1)}></button>;  
}
```

- ☐ a. Increments after every click
- ☒ b. Always 0
- ☐ c. The latest count value
- ☐ d. A different number each second

**Question 3**

Complete

Mark 1.00 out of 1.00

What is printed to the console on the first button click?

```
function App() {  
  const [state, setState] = React.useState(0);  
  const ref = React.useRef(0);  
  const handleClick = () => {  
    ref.current += 1;  
    setState(state + 1);  
    console.log("State:", state, "Ref:", ref.current);  
  };  
}
```

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

- ☐ a. State: 1 Ref: 0
- ☒ b. State: 0 Ref: 1
- ☐ c. State: 1 Ref: 2
- ☐ d. State: 1 Ref: 1

**Question 4**

Complete

Mark 1.00 out of 1.00

What will be logged to the console when the following component is rendered inside `<React.StrictMode>`?

```
function App() {  
  React.useEffect(() => {  
    console.log("Effect ran");  
  }, []);  
}
```

```
  return <div>Hello</div>;  
}
```

- ☐ a. Effect ran
- ☐ b. Compilation error
- ☐ c. Nothing
- ☒ d. Effect ran (logged twice)

**Question 5**

Complete

Mark 1.00 out of 1.00

Why does the following component cause an infinite render loop?

```
function App() {  
  const [count, setCount] = React.useState(0);  
  const obj = {  
    increment: () => setCount(count + 1),  
  };  
  React.useEffect(() => {  
    obj.increment();  
  }, [obj]);  
}
```

```
  return <div>{count}</div>;  
}
```

- ☐ a. Because count changes inside useEffect
- ☒ b. Because obj is re-created on every render
- ☐ c. There is no loop
- ☐ d. Because increment modifies state incorrectly