

**Started on** Tuesday, 27 May 2025, 2:39 PM

**State** Finished

**Completed on** Tuesday, 27 May 2025, 2:52 PM

**Time taken** 12 mins 57 secs

**Marks** 5.00/5.00

**Grade** 100.00 out of 100.00

## Question 1

Complete

Mark 1.00 out of 1.00

Flag question

### Question text

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>;  
}
```

### Question 1 Answer

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

## Question 2

Complete

Mark 1.00 out of 1.00

Flag question

### Question text

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>;  
}
```

#### Question 2 Answer

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

### Question 3

Complete

Mark 1.00 out of 1.00

Flag question

#### Question text

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>;  
}
```

#### Question 3 Answer

- ☐ a.  
State: 1 Ref: 0
- ☐ b.  
State: 1 Ref: 1



c.

State: 1 Ref: 2



d.

State: 0 Ref: 1

## Question 4

Complete

Mark 1.00 out of 1.00

Flag question

### Question text

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>;  
}
```

### Question 4 Answer



a.

Effect ran



b.

Compilation error



c.

Effect ran (logged twice)



d.

Nothing

## Question 5

Complete

Mark 1.00 out of 1.00

Flag question

### Question text

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>;  
}
```

#### Question 5 Answer

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