

Started on	Tuesday, 5 August 2025, 4:36 PM
State	Finished
Completed on	Tuesday, 5 August 2025, 4:50 PM
Time taken	13 mins 12 secs
Marks	13.00/20.00
Grade	65.00 out of 100.00

Question 1

Complete

Mark 0.00 out of 1.00

What will be logged to the console in the following code?

```
const foo = () => {  
  return {  
    bar: function () {  
      console.log(this);  
    }  
  };  
};  
  
const obj = foo();  
obj.bar();
```

- ☐ a. `undefined`
- ☐ b. `global` (or `window` in the browser)
- ☒ c. `null`
- ☐ d. `{ bar: function () { console.log(this); } }`

Question 2

Complete

Mark 0.00 out of 1.00

Which of the following methods can be used to create a shallow copy of an array?

- ☐ a. `Object.assign()`
- ☐ b. `JSON.parse()`
- ☐ c. `Array.slice()`
- ☒ d. `Array.copy()`

Question 3

Complete

Mark 1.00 out of 1.00

Which of the following statements is true about JavaScript closures?

- ☐ a. Closures do not retain access to variables declared in outer scopes.
- ☐ b. Closures prevent memory leaks.
- ☐ c. Closures are only created when functions are called, not when they are defined.
- ☒ d. A closure is created when a function is defined within another function, and the inner function gains access to the outer function's variables.

Question 4

Complete

Mark 0.00 out of 1.00

What is the output of this code?

```
function foo() {  
  console.log(this);  
}  
const obj = { foo: foo };  
obj.foo();
```

- ☒ a. `undefined`
- ☐ b. `null`
- ☐ c. `window`
- ☐ d. `obj`

Question 5

Complete

Mark 1.00 out of 1.00

What is the result of the following code?

```
function factorial(n) {  
  return n <= 1 ? 1 : n * factorial(n - 1);  
}  
console.log(factorial(5));
```

- ☐ a. `24`
- ☐ b. `5`
- ☒ c. `120`
- ☐ d. `Error`

Question 6

Complete

Mark 1.00 out of 1.00

What will be the output of the following code?

```
const arr = [1, 2, 3];  
const newArr = [...arr, 4];  
console.log(newArr);
```

- ☐ a. `[1, 4, 2, 3]`
- ☐ b. `[4, 1, 2, 3]`
- ☐ c. `[1, 2, 3]`
- ☒ d. `[1, 2, 3, 4]`

Question 7

Complete

Mark 1.00 out of 1.00

What will be the output of the following code?

```
let x = 10;
function example() {
  console.log(x);
  if (true) {
    console.log(x);
    let x = 20;
  }
}
example();
```

- ☒ a. `ReferenceError`
- ☐ b. `10 20`
- ☐ c. `undefined undefined`
- ☐ d. `undefined 20`

Question 8

Complete

Mark 1.00 out of 1.00

What is the output of the following code?

```
function asyncFunc() {
  console.log(1);
  setTimeout(() => console.log(2), 0);
  console.log(3);
}
asyncFunc();
```

- ☒ a. `1 3 2`
- ☐ b. `1 2 3 4`
- ☐ c. `1 3`
- ☐ d. `1 2 3`

Question 9

Complete

Mark 1.00 out of 1.00

What will be logged to the console?

```
let x = 5;
let y = 10;
if (x === 5) {
  let y = 15;
  console.log(y);
}
console.log(y);
```

- ☐ a. `15 15`
- ☒ b. `15 10`
- ☐ c. `5 10`
- ☐ d. `Error`

Question 10

Complete

Mark 1.00 out of 1.00

What will the following code output?

```
let x = 5;
function foo() {
  x = 10;
  return;
  function x() {}
}
foo();
console.log(x);
```

- ☐ a. `10`
- ☒ b. `5`
- ☐ c. `undefined`
- ☐ d. `ReferenceError`

Question 11

Complete

Mark 1.00 out of 1.00

What will the following code output?

```
let x = { a: 1 };
let y = { a: 1 };
console.log(x == y);
```

- ☐ a. `Error`
- ☒ b. `false`
- ☐ c. `undefined`
- ☐ d. `true`

Question 12

Complete

Mark 0.00 out of 1.00

What will be logged to the console in the following code?

```
let a = 10;
let b = 20;
function foo() {
  console.log(a);
  return;
  let a = 30;
}
foo();
```

- ☒ a. ReferenceError
- ☐ b. 30
- ☐ c. 10
- ☐ d. undefined

Question 13

Complete

Mark 1.00 out of 1.00

What will be the output of the following code?

```
const foo = async () => {  
  return 'Hello';  
};  
  
const bar = foo();  
console.log(bar);
```

- ☐ a. `Error`
- ☒ b. `Promise { 'Hello' }`
- ☐ c. `undefined`
- ☐ d. `Hello`

Question 14

Complete

Mark 1.00 out of 1.00

What is the result of the following code?

```
const clone = [1, 2, 3];  
clone.push(4);  
console.log(clone);
```

- ☐ a. `[4, 1, 2, 3]`
- ☒ b. `[1, 2, 3, 4]`
- ☐ c. `[1, 2, 3]`
- ☐ d. `Error`

Question 15

Complete

Mark 0.00 out of 1.00

What will be the output of the following code?

```
let a = 10;  
let b = 20;  
function test() {  
  console.log(this.a + this.b);  
}  
test();
```

- ☐ a. `NaN`
- ☒ b. `Error`
- ☐ c. `30`
- ☐ d. `undefined`

Question 16

Complete

Mark 1.00 out of 1.00

Which of the following is true about JavaScript's event loop?

- ☒ a. It executes all synchronous code before any asynchronous code.
- ☐ b. It only processes synchronous code.
- ☐ c. It executes both synchronous and asynchronous code in parallel.
- ☐ d. It executes all asynchronous code before any synchronous code.

Question 17

Complete

Mark 1.00 out of 1.00

What will the following code output?

```
const arr = [1, 2, 3];  
arr[10] = 11;  
console.log(arr);
```

- ☒ a. `[1, 2, 3, 11]`
- ☐ b. `Error`
- ☐ c. `[1, 2, 3]`
- ☐ d. `[4, 1, 2, 3]`

Question 18

Complete

Mark 0.00 out of 1.00

What is the result of the following code?

```
let x = 5;  
function increment() {  
  x++;  
  return x;  
}  
increment();  
increment();  
increment();  
console.log(x);
```

- ☐ a. `6`
- ☐ b. `7`
- ☐ c. `5`
- ☒ d. `8`

Question 19

Complete

Mark 0.00 out of 1.00

What will be the output of the following code?

```
let a = 10;  
let b = a;  
b = 20;  
console.log(a);  
console.log(b);
```

- ☐ a. `undefined undefined`
- ☒ b. `20 20`
- ☐ c. `10 10`
- ☐ d. `10 20`

Question 20

Complete

Mark 1.00 out of 1.00

Which of the following will correctly handle asynchronous operations in JavaScript?

- ☒ a. Using promises and callbacks.
- ☐ b. Using synchronous functions only.
- ☐ c. Using `setTimeout` and `setInterval` exclusively.
- ☐ d. Using loops to handle async code.