

| | |
|---------------------|---------------------------------|
| Started on | Tuesday, 5 August 2025, 4:37 PM |
| State | Finished |
| Completed on | Tuesday, 5 August 2025, 4:49 PM |
| Time taken | 12 mins 47 secs |
| Marks | 16.00/20.00 |
| Grade | 80.00 out of 100.00 |

Question 1

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

Question 2

Complete

Mark 1.00 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. `20 20`
- ☐ b. `10 10`
- ☐ c. `undefined undefined`
- ☒ d. `10 20`

Question 3

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

Question 4

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 5

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

Question 6

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

Question 7

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

Question 8

Complete

Mark 1.00 out of 1.00

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

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

Question 9

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

Question 10

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

Question 11

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

Question 12

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

Question 13

Complete

Mark 1.00 out of 1.00

What is the output of this code?

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

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

Question 14

Complete

Mark 1.00 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. `30`
- ☐ b. `Error`
- ☐ c. `undefined`
- ☒ d. `NaN`

Question 15

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

Question 16

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. `Error`
- ☒ c. `120`
- ☐ d. `5`

Question 17

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

Question 18

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

Question 19

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

Question 20

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