1 . What is a closure in JavaScript?

A) A function that returns another function

B) A function bundled with its lexical environment

C) A function that calls itself

D) A function with no parameters

3. Closures are created when:

A) A function is defined inside another function

B) A variable is declared globally

C) A function is called recursively

D) A function is assigned to a variable

4. What will the following code output?

function **makeCounter**() {

  let count = 0;

  return function() {

    count++;

    return count;

  };

}

const **counter** = **makeCounter**();

console.**log**(**counter**());

console.**log**(**counter**());

A) 0, 0

B) 1, 2

C) 1, 1

D) 2, 2

5. Closures can be used to:

A) Create private variables

B) Create global variables

C) Prevent function execution

D) Increase performance

6. What is the value of result?

function **outer**() {

  let x = 5;

  function **inner**() {

    return x \* 2;

  }

  return inner;

}

const result = **outer**()();

A) 5

B) 10

C) undefined

D) Error

7. Which statement is true about closures?

A) Closures can access variables from their parent scope

B) Closures cannot access global variables

C) Closures only work with arrow functions

D) Closures are only available in ES6

8. What will this code output?

function **test**() {

  let arr = [];

  for (var i = 0; i < 3; i++) {

    arr.**push**(function() { return i; });

  }

  return arr;

}

const result = **test**();

console.**log**(result[0]());

console.**log**(result[1]());

console.**log**(result[2]());

A) 0, 1, 2

B) 3, 3, 3

C) 2, 2, 2

D) 1, 2, 3

9. How can you fix the previous code to output 0, 1, 2?

A) Use let instead of var for i

B) Use const instead of var for i

C) Remove the loop

D) Use global variables

10. Closures are commonly used for:

A) Event handlers

B) Callback functions

C) Data encapsulation

D) All of the above

1. **B** – A function bundled with its lexical environment
2. **A**
3. **B** – 1, 2
4. **A** – Create private variables
5. **B** – 10
6. **A**
7. **B** – 3, 3, 3
8. **A** – Use let instead of var for i
9. **D** – All of the above