Question 1

What will be the output of the below code:

```
const array = [10, 20, 30, 40];
const result = array.map((num) => num / 2).filter((num) => num >= 15);
console.log(result);
```

Question 2

Find the issue in the below code snippet.

```
 let counter = 0;
for (var i = 1; i <= 10; i++) {
  counter+= i;
}
console.log(counter);
console.log(i);
```

Question 3

Analyze the below code. Do you see any issue? If yes, what is that issue?

```
const object1 = {
  a: 10,
  b: 20,
  c: function () {
        console.log(this.a + this.b);
  },
};
const func = object1.c;
func();
```

Question 4

Create a JavaScript function that calculates the tip for a given bill amount and tip percentage. Bill amount and tip percentage will be input parameters while output will be calculated tip value.

UNCODEMY

Question 5

What will be the output of below code snippet:

```
function greetHello(name) {
  return `Hello, ${name}!`;
}
console.log(greetHello("Brian"));
```

Question 6

Will the below code return any error? If yes, identify the error.

```
function fetchData(callback) {
  fetch('https://api.example.com/data')
        .then(response => response.json())
        .then(data => callback(null, data))
        .catch(error => callback(error));
}
 fetchData(function (error, data) {
  if (error) {
        console.log('Error:', error);
  } else {
        console.log('Data:', data);
  }
});
```

UNCODEMY

Question 7

Implement a simple shopping cart system with features to add items, remove items and calculate the total price. Use objects to represent items, including properties for the item name, price and quantity. Implement features to add items to the cart, remove items and calculate the total cost.

Question 8

Analyze the below code snippet and advise what will be the output:

```
 const person = {
```

```
  firstName: "Helen",
  lastName: "Ryan",
  getFullName: function () {
        return this.firstName + " " + this.lastName;
  },
};
console.log(person.getFullName());
```

Question 9

Find the issue with the below code snippet:

```
setTimeout(function () {
  console.log("This will be executed after 3 seconds");
}, 3000);
clearTimeout();
```

Question 10

What issue exists in the below code:

```
const testArray = [1, 2, 3];
testArray = [4, 5, 6];
console.log(testArray);
```

UNCODEMY

Interview Questions for Mid-level JavaScript Developers

Question 1

What is the issue in the below code:

```
 const fetchData = async () => {
  const response = await fetch("https://api.samplewebsite.com/data");
  const data = await response.json();
  console.log(data);
};
fetchData();
```

Question 2

What will be the output of the below code:

```
const promise1 = Promise.resolve("One");
const promise2 = new Promise((resolve) => setTimeout(() =>
resolve("Two"), 1000));
const promise3 = Promise.reject("Three");
 Promise.allSettled([promise1, promise2, promise3]).then((results) =>
console.log(results));
```

Question 3

Develop a simple URL shortener service using JavaScript. Implement a function that takes a long URL as an input parameter and the output will be a shortened URL. Create a reverse function as well. The reverse function takes the shortened URL and returns the original long URL. You can use simple in-memory objects to store the mapping between long and short URLs.

Question 4

Implement an autocomplete feature for a search input field. Given an array of words, write a function that suggests words based on the current input. The output of the function will be an array of suggested words that start with the input characters, limiting the number of suggestions (e.g., a maximum of 7 suggestions).

Question 5

What is the issue in the below code:

```
const obj = {
  name: "Conner",
  age: 27,
  greet: () => {
        console.log(`Hey, my name is ${this.name}`);
  },
};
```

UNCODEMY

```
obj.greet();
```

Question 6

What will be the output of below code snippet:

```
const object1 = {
  prop1: "value1",
  prop2: {
        prop3: "value3",
  },
};
 const newObj = { ...obj };
newObj.prop2.prop3 = "newValue3";
console.log(object1.prop2.prop3);
```

Question 7

Will the below code return any error? If yes, what will be the error?

```
class Bird {
  constructor(name) {
        this.name = name;
  }
  speak() {
        console.log(`${this.name} makes a noise.`);
  }
}
 class Crow extends Bird{
  speak() {
        super.speak();
        console.log(`${this.name} sings.`);
  }
}
 const crow = new Crow("Tim");
crow.speak();
```

Question 8

UNCODEMY

Develop a function that throttles another function, allowing it to be called at most once every specified interval (e.g., 300ms). The throttling function will have two input parameters. One will be the function to be throttled and the second will be the interval in milliseconds. The throttled function should be called with the same arguments as the original function.

Question 9

What is wrong with the below code:

```
const arr = [1, 2, 3, 4, 5];
const sum = arr.reduce((total, num) => total + num);
console.log(sum / arr.length);
```

Question 10

Design a simple meeting scheduler that finds the first available time slot for a meeting between two people. Given two arrays of busy time intervals and a meeting duration, create a function that returns the earliest available time slot for the meeting when both people will be available. Each interval is represented as an array of two integers, where the first integer is the start time and the second integer is the end time.

UNCODEMY

Interview Questions for Expert JavaScript Developers

When preparing coding challenges for expert-level JavaScript engineers, you should test the advanced features of the language and performance optimization techniques. Some of the areas to evaluate include advanced JavaScript features, code architecture, design patterns, performance optimization and security. Below we have presented 10 coding challenges for expert JavaScript developers:

Question 1

Is there any security vulnerability in the below code? If yes, identify it:

```
const username = document.getElementById('username').value;
const password = document.getElementById('password').value;
fetch('https://api.examplewebsite.com/login', {
  method: 'POST',
  body: JSON.stringify({ username, password })
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.log(error));
```

Question 2

Identify the output of the below code.

```
const testArray = [1, 2, 3, 4, 5];
const res = testArray.reduce((acc, curr) => {
  if (curr % 2 === 0) {
        return acc + curr;
  }
  return acc;
}, 0);
console.log(res);
```

UNCODEMY

Question 3

What is the possible performance issue in the below code?

```
const arr = [];
for (let i = 0; i < 1000000; i++) {
  arr.push(Math.floor(Math.random() * 1000));
}
```

Question 4

Suggest the output of the below code:

```
const arr = [1, 2, 3];
const object1 = { x: 1, y: 2, z: 3 };
```

```
console.log([...arr, ... object1]);
```

Question 5

Design a social media platform that contains features like sign up, creating a profile and posting status updates. Users should be able to follow other users and view their posts on a newsfeed.

Question 6

What is wrong with the below call to the API?

```
fetch('https://api.example.com/data')
  .then(response => {
        if (!response.ok) {
        throw new Error('Network response was not ok');
        }
        return response.json();
  })
  .then(data => console.log(data))
  .catch(error => console.log(error));
```

UNCODEMY

Question 7

What will be the output of below code snippet?

```
const promise1 = Promise.resolve(One);
const promise2 = Promise.resolve(Two);
Promise.all([promise1, promise2]).then(([result1, result2]) =>
console.log(result1 + ' ' + result2));
```

Question 8

Design an online code editor where users can write, save and run JavaScript code. The editor should include features like syntax highlighting, auto-completion and error checking.

Question 9

The below code snippet uses closures to implement a counter. How will you optimize it to minimize memory usage:

```
 function counter() {
  let count = 0;
  return function() {
        count++;
        console.log(count);
  }
}
const increment = counter();
increment(); // 1
increment(); // 2
increment(); // 3
```

UNCODEMY