# Answersheet

Name: Arashad Ahamad
No. of questions attempted: 11
Submitted at: 18 Jun, 2025 7:41 PM

Assignment: Map, Filter, Reduce in JavaScript
Total no. of questions: 11
Total marks: 39

Result: Not reviewed yet

| Sl. No. | Question | Actions |
|---|---|---|
| 1 | **How can you access the index of each element inside the "forEach" and "map" methods in JavaScript?** 1 mark<br>1. By using the Array.prototype.indexOf() method.<br>2. By using a global counter variable.<br>3. **By using the second argument of the callback function.**<br>4. By using the Array.prototype.indexOf() method. | **Correct answer**<br>1 mark |

| | | |
|---|---|---|
| 2 | **What is the primary purpose of the "filter" method in JavaScript?** 1 mark<br>1. To transform each element of an array based on a callback function<br>2. To reduce an array to a single value based on a callback function<br>3. **To create a new array with only elements that pass a certain condition**<br>4. To sort the elements of an array in ascending order | **Correct answer**<br>1 mark |
| 3 | **How does chaining of array methods work in JavaScript?** 1 mark<br>1. **It allows you to combine multiple array methods together to perform complex operations**<br>2. It restricts the usage of array methods to only one method per array<br>3. It ensures that array methods cannot be used together<br>4. It enforces a specific order in which array methods can be used | **Correct answer**<br>1 mark |
| 4 | **What does the "reduce" method do in JavaScript?** 1 mark<br>1. It creates a new array by selecting specific elements from the original array<br>2. **It applies a function against an accumulator and each element in the array to reduce it to a single value**<br>3. It removes elements from the array that do not satisfy a given condition<br>4. It sorts the elements of the array in ascending order | **Correct answer**<br>1 mark |

| | | |
|---|---|---|
| 5 | **Given an array of numbers, create a new array that contains the squares of each number.** 5 marks<br>***const numbers = [1, 2, 3, 4, 5];***<br>***// Your code here***<br>***// Expected Output: [1, 4, 9, 16, 25]***<br>const numbers8 = [1, 2, 3, 4, 5] const squareNumbers2 = numbers8.map((num) => { return num * num }) console.log(squareNumbers2); | Marks<br>0 |
| 6 | **Given an array of numbers, filter out the even numbers and return a new array with only odd numbers.** 5 marks<br>***const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];***<br>***// Your code here***<br>***// Expected Output: [1, 3, 5, 7, 9]***<br>const numbers9 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; const oddNumber = numbers9.filter((num) => { return num % 2 !== 0 }) console.log(oddNumber); | Marks<br>0 |
| 7 | **Given an array of numbers, find the product of all the elements in the array.** 5 marks<br>***const numbers = [1, 2, 3, 4, 5];***<br>***// Your code here***<br>***// Expected Output: 120*** | Marks<br>0 |

const numbers10 = [1, 2, 3, 4, 5] const multiply2 = numbers10.reduce((acc, num) =>{ return acc * num }, 1) console.log(multiply2);

| | | | |
|---|---|---|---|
| 8 | **Given an array of numbers, create a new array that contains the squares of only the odd numbers.**<br>***const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];***<br>***// Your code here***<br>***// Expected Output: [1, 9, 25, 49, 81]***<br>const numbers11 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; const oddNumberSquares = numbers11.filter((num) => { return num % 2 !== 0 }).map(num => num * num) console.log(oddNumberSquares); | 5 marks | Marks<br>0 |
| 9 | **Given an array of strings, count the number of times each string appears and return an object with the string as the key and the count as the value.**<br>***const fruits = ['apple', 'banana', 'apple', 'orange', 'banana', 'apple'];***<br>***// Your code here***<br>***// Expected Output: { apple: 3, banana: 2, orange: 1 }***<br>const fruits3 = ['apple', 'banana', 'apple', 'orange', 'banana', 'apple']; const count = fruits3.reduce((acc, fruit) => { acc[fruit] = (acc[fruit] || 0) + 1 return acc }, {}) console.log(count) | 5 marks | Marks<br>0 |
| 10 | **Given an array of objects representing products with a price property, calculate the total cost of all products after applying a 10% discount.** | 5 marks | Marks |

***const products = [***

***{ name: 'Product 1', price: 100 },***

***{ name: 'Product 2', price: 200 },***

***{ name: 'Product 3', price: 300 }***

***];***

***// Your code here***

***//Expected Output: 540***

const products = [ { name: 'Product 1', price: 100 }, { name: 'Product 2', price: 200 }, { name: 'Product 3', price: 300 } ]; const totalCost = products.reduce((acc, product) => { const discountedPrice = product.price - (product.price * 10) / 100; return acc + discountedPrice; }, 0); console.log(totalCost);

0

11

**Given three arrays, names, ages, and cities, write a program to combine them into one array of objects where each object contains a name, age, and city property.**

5 marks

***const names = ['Alice', 'Bob', 'Charlie'];***

***const ages = [25, 30, 35];***

***const cities = ['New York', 'Los Angeles', 'Chicago'];***

***// Your code here***

***// Expected Output: [***

***// { name: 'Alice', age: 25, city: 'New York' },***

***// { name: 'Bob', age: 30, city: 'Los Angeles' },***

Marks

0

*// { name: `Charlie`, age: 35, city: `Chicago` }*

*// ]*

const names = ['Alice', 'Bob', 'Charlie']; const ages = [25, 30, 35]; const cities = ['New York', 'Los Angeles', 'Chicago']; const combined = names.map((name, index) => { return { name: name, age: ages[index], city: cities[index] }; }); console.log(combined);