Javascript Assignment 8

1. Can we put duplicate values in the set object ?

Ans:

No, you cannot put duplicate values in a Set object in JavaScript. A Set is a collection of unique values, and it automatically enforces uniqueness. If you attempt to add a duplicate value to a Set, it will be ignored, and the Set will continue to contain only unique values.

const mySet = new Set();

mySet.add(1);

mySet.add(2);

mySet.add(3);

mySet.add(1); // This duplicate value will be ignored

console.log(mySet); // Output: Set { 1, 2, 3 }

2. Write the syntax for

a) Creating new set object

b) Adding new element to set object

c) Deleting element from set object

Ans:

1. Creating a new Set object:

const mySet = new Set();

1. Adding a new element to a Set object:

mySet.add(element);

1. Deleting an element from a Set object:

mySet.delete(element);

Example:

const mySet = new Set();

mySet.add(1);

mySet.add(2);

mySet.delete(1); // Removes the value 1 from the Set

3. Create a set object with four random numbers from 0 to 10. Check if this

newly created set object has 8 in it. Use set object methods.

Ans:

// Function to generate a random integer between min and max (inclusive)

function getRandomInt(min, max) {

return Math.floor(Math.random() \* (max - min + 1)) + min;

}

// Create a Set with four random numbers

const mySet = new Set();

while (mySet.size < 4) {

const randomNumber = getRandomInt(0, 10);

mySet.add(randomNumber);

}

console.log("Set:", mySet);

// Check if the Set contains the number 8

const hasEight = mySet.has(8);

if (hasEight) {

console.log("The Set contains the number 8.");

} else {

console.log("The Set does not contain the number 8.");

}