Javascript Assignment 6

1. Perform the following operations to provide the implementation for a

Rectangle class. The operations are:

1. Add an area() method to the Rectangle class.

2. Create a Square class that satisfies the following conditions:

○ It is a subclass of Rectangle.

○ It contains a constructor and no other methods.

○ It can use the Rectangle class' area method to print the area

of a Square object.

Ans:

class Rectangle {

constructor(width, height) {

this.width = width;

this.height = height;

}

area() {

return this.width \* this.height;

}

}

class Square extends Rectangle {

constructor(side) {

super(side, side); // Call the superclass (Rectangle) constructor with the same width and height

}

}

// Example usage:

const rectangle = new Rectangle(4, 5);

console.log(`Rectangle Area: ${rectangle.area()}`); // Output: Rectangle Area: 20

const square = new Square(4);

console.log(`Square Area: ${square.area()}`); // Output: Square Area: 16

2. Write a javascript function find\_largest to return the nth largest number

in an array-

eg- given an array of integers- [3,45,6,7,23,5,7,8]

find\_largest(3) will return third largest number from the above array -

which is 8.

Ans:

function find\_largest(arr, n) {

if (n <= 0 || n > arr.length) {

return "Invalid input";

}

// Sort the array in descending order

arr.sort((a, b) => b - a);

// Return the nth largest number

return arr[n - 1];

}

// Example usage:

const numbers = [3, 45, 6, 7, 23, 5, 7, 8];

const nthLargest = find\_largest(numbers, 3);

console.log(`The 3rd largest number is: ${nthLargest}`); // Should print 8

3. Write a JavaScript program which accept a number as input in the

function parameter and insert dashes (-) between each two even

numbers.

For example if you accept 025468 as the output should be 0-254-6-8.

computeDash(025468) -> 0-254-6-8.

Ans:

function computeDash(inputNumber) {

const numberStr = inputNumber.toString();

let result = '';

for (let i = 0; i < numberStr.length; i++) {

const currentDigit = parseInt(numberStr[i]);

const nextDigit = parseInt(numberStr[i + 1]);

result += currentDigit;

if (currentDigit % 2 === 0 && nextDigit % 2 === 0) {

result += '-';

}

}

return result;

}

// Example usage:

const inputNumber = 025468;

const dashedNumber = computeDash(inputNumber);

console.log(dashedNumber); // Should print "0-254-6-8"