

LAB NO.

\_\_\_\_7\_\_\_

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Write a javascript function that takes length, width, and height values of rectangle from user. The function should find the volume of rectangle using the function-currying. |
| 2 | Create two functions “Profile” (outer function) and “greetingMsg” (inner function) that aims to implements the concept of Function-closure by generating an alert of greeting message with the help of user details provided in outer function. |
| 3 | It's a general concept in mathematics where you combine two or more functions into a brand-new function. Write a javascript program to implement the given concept with the help of function-compose for the given function. f(g(x)) |
| 4 | Write a javascript program that uses  filter() to create a filtered array that has all elements with values less than 10 removed. |
| 5 | Creates an array consisting of only those elements that satisfy the condition checked by isPositive() function with the help of appropriate javascript advance loops concept. |
| 6 | Write a javascript program that implements the array.**map()** that aim to produces an array containing square roots of the numbers in the original array. |
| 7 | Create a class named 'Member' having the members: Name, Age, Salary. It also has a method named 'printSalary' which prints the salary of the members. Create child class 'Employee' that inherits the 'Member' class. The 'Employee' classes have data member 'department'. Now, assign name, age and salary to an employee by making an object of child class and print the same. |
| 8 | Write a javascript program to implement the concept of nullish coalesing operator by using the below object properties.  const response = { data: { name: 'Ronaldo’, occupation: null, lies: 0 } } |

**Lab Task**

1. Write a javascript function that takes length, width, and height values of rectangle from user. The function should find the volume of rectangle using the function-currying.

CODE:

function curriedVolume(l) {

return (w) => {

return (h) => {

return l \* w\* h

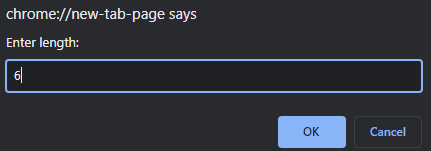
}}} var length=prompt("Enter length: ");

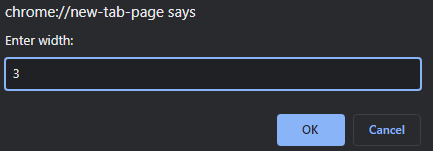
var width=prompt("Enter width: ");

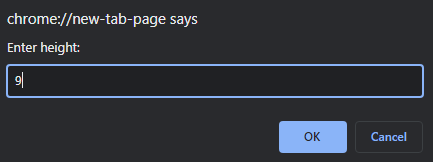
var height=prompt("Enter height: ")

console.log(curriedVolume(length)(width)(height));

Output: 162







1. Create two functions “Profile” (outer function) and “greetingMsg” (inner function) that aims to implements the concept of Function-closure by generating an alert of greeting message with the help of user details provided in outer function.

function Profile() {

    var message = "Hello Mr. Smith! Hope you have a wonderful day";

    function greetingMsg() {

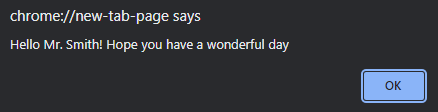
        alert(message);

    }

    return greetingMsg;}

var greetingMsg = Profile();

greetingMsg();



1. It's a general concept in mathematics where you combine two or more functions into a brand-new function. Write a javascript program to implement the given concept with the help of function-compose for the given function. **f(g(x))**

const compose = (disOnBill,Total) => tax => disOnBill(Total(tax));

const Total = (x) => x + 500 //tax

const disOnBill = (x) => x \* 0.75 //25%

const BillPaid = compose(disOnBill,Total)

BillPaid(25000)

Output: 19125

1. Write a javascript program that uses  filter() to create a filtered array that has all elements with values less than 10 removed.



const array = [ 1, 3, 7, 5, 11,15,17,19,21];

const filterArray = array.filter(num => num > 10);

console.log(filterArray);

1. Creates an array consisting of only those elements that satisfy the condition checked by **isPositive()** function with the help of appropriate javascript advance loops concept.

const numbers = [175, -50, 25];

numbers.filter(myFunc);

function myFunc(num) {

return num > 0;

}

Output: [175, 25];

1. Write a javascript program that implements the array.**map()** that aim to produces an array containing square roots of the numbers in the original array.

const numbers = [1, 2, 3, 4, 5];

const bigNumbers = numbers.map(number => {

return number \* 10;

});

bigNumbers

Output:  [10, 20, 30, 40, 50]

1. Create a class named 'Member' having the members: Name, Age, Salary. It also has a method named 'printSalary' which prints the salary of the members. Create child class 'Employee' that inherits the 'Member' class. The 'Employee' classes have data member 'department'. Now, assign name, age and salary to an employee by making an object of child class and print the same.

class Member{

constructor (Name, Age, Salary){

this.Name=Name;

this.Age=Age;

this.Salary=Salary;

}

printSalary(){

console.log(Salary);

}

}

class Employee extends Member{

constructor(Name, Age, Salary,department){

super(Name, Age, Salary);

this.department=department;

}

}

const a = new Employee("ALI", 21, 25000, "abc");

undefined

a.printSalary

1. Write a javascript program to implement the concept of nullish coalesing operator by using the below object properties.

const response = {  
data: {  
name: 'Ronaldo’,  
occupation: null,  
lies: 0  
}  
}

const response = {

data: {

name: 'Ronaldo’,

occupation: null,

lies: 0

}

}

const occupation2 = response?.data?.occupation || 'no occupation' ;

console.log(occupation2);

Output: no occupation