ES6 Assignments

1. **Constants:** Declare a constant & confirm its value cannot be changed.

Ans:

const num = 9;

num = 5;

console.log(num);

1. **Scoping:** Declare a variable inside if condition & make sure that it is not accessible outside if condition.

function testNum(a)

{

    let result;

    if(a>0)

    {

        result = 'Positive';

    }

    else

    {

        result = 'Not Positive';

    }

    return result;

}

//console.log(result);

console.log(testNum(-5));

1. **Enhanced object properties:** Create an ‘Order’ object having data members ‘id’, ‘title’, ‘price’. Add the methods printOrder() & getPrice(). Now, copy the order object using Object.assign().

var Order =

{

    id:1,

    title:"Pizza",

    price:100,

    // var data = { id:1, title:"Pizza",price:100};

    printOrder:function()

    {

        console.log(this.id);

        console.log(this.title);

        console.log("The order is "+ " "+ this.title);

    },

    getprice:function()

    {

        console.log("The price is "+ " "+ this.price);

    }

}

var O1 = Object.create(Order);

O1.printOrder();

O1.getprice();

var copy = Object.assign({},Order);

for(let val in copy)

{

    console.log(copy[val]);

}

1. **Arrow functions:** Take an array of strings & convert it into another array of object which has two properties {string, string\_length}. For example:

let names = [‘Tom’, ‘Ivan’, ‘Jerry’]

Output: [ {name: ’Tom’, length: 3}, {name: ’Ivan’, length: 4 }, {name: ’Jerry’, length: 5} ]

function give()

{

    let name = ["Tom","Ivan","Jerry"];

    var output = [];

    for(var i = 0;i<name.length;i++)

    {

        output[i] = {

            name: name[i],

            length: name[i].length

        };

    }

    console.log(output);

}

give();

1. **Extended parameter handling:**
   1. Write a add() with default values.
   2. Write a function userFriends() that takes 2 arguments username & array of user friends. The function should print username & his list of friends. (Use rest parameters)
   3. Write a function printCapitalNames() that takes five names as argument & prints them in capital letters. Use spread operator in order to call printCapitalNames() function.

let add = (a = 2,b = 4) => {

    console.log(a+b);

}

add();

let userFriends = function(name, ...friends)

{

    console.log(name);

    for(let i in friends)

    {

        console.log(friends[i]);

    }

}

userFriends("Bhushan", 'anant','dada','upa');

function printCapitalNames(...name)

{

    for(let i in name)

    {

        console.log(name[i].toUpperCase());

    }

}

let name = ["sa","da","fd","gf","bf"];

(printCapitalNames(...name));

1. **Template literals:** Draft a ticket to Sysnet that describes problem with your laptop. Use ‘template literals’ to add value of laptop model, your desk no, your name etc.

let name = "Bhushan";

let deskno = 20;

let model = "XPS-15";

console.log(`${name}` + ` `+ `deskno`+` `+ `model`);

1. **De-structuring assignment:**
   1. Suppose there is a javascript array with 4 elements. Print the value of 3rd element using array matching.

const foo = ["A","B","C","D"];

const [one,two,three,four] = foo;

console.log(three);

* 1. Create an organization object having attributes name, address. Write a program to retrieve pin code of an address using object deep matching.

1. **Classes & Modules:** Write a class Account with attributes id, name, balance. Add two sub classes SavingAccount & CurrentAccount having specific attribute interest & cash\_credit respectively. Create multiple saving & current account objects. Write a functionality to find out total balance in the bank.

class Account

{

    constructor(id,name,balance)

    {

        this.id = id;

        this.name = name;

        this.balance = balance;

    }

    tbal()

    {

        console.log("The balance is" + " "+this.balance);

    }

}

class SavingAccount extends Account

{

    constructor(id,name,balance,interest)

    {

        super(id,name,balance);

        this.interest = interest;

    }

    tbalsave()

    {

        console.log("The balance is" + " "+this.interest+" "+this.balance);

    }

}

class CurrentAccount extends Account

{

    constructor(id,name,balance,cash\_credit)

    {

        super(id,name,balance);

        this.cash\_credit = cash\_credit;

    }

    tbalcur()

    {

        console.log("The balance is" + " "+this.balance+" "+ this.cash\_credit );

    }

}

let a1 = new SavingAccount(1,"Anant",12000,300);

let a2 = new CurrentAccount(2,"Basant",1200,100);

let a3 = new Account(3,"Joe",4586);

a1.tbalsave();

a2.tbalcur();

a3.tbal();