1. Write a write up on difference between copy by value and copy by reference.

Ans: Copy by value:

In a primitive datatype when a variable is assigned a value we can imagine that a box is created in the memory. This box has a sticker attached to it i.e. the variable name. Inside the box the value assigned to the variable is stored.

Example: **Variables are copied**

var x= 5;

var y = ‘xyx’;

var z = null;

var a = x;

var b = y;

console.log(x,y,a,b);

o/p: 5, ‘xyz’ , 5 , ‘xyz’.

At this point of time both ‘x’ and ‘a’ contains the value 5. Both ‘y’ and ‘b’ contains the value ‘xyz’. However, an important thing to understand here is that even though ‘x’ and ‘a’ as well as ‘y’ and ‘b’ contains the same value they are not connected to each other. It is so because the values are directly copied into the new variables.

Changes taking place in one does not affect the other.

Example: **Variables are independent of each other**

var x = 5;

var y = ‘xyz’;

var z = null;

var a = x;

var b = y;

x = 7;

y = ‘das’;

console.log(x, y, a, b);

o/p:5 , ‘xyz’ , 7, 5, ‘xyz’.

Copy by reference

In case of a non-primitive datatype or composite datatypes the values are not directly copied. When a non-primitive datatype is assigned a value a box is created with a sticker of the name of the data-type. However, the values it is assigned is not stored directly in the box. The language itself assigns a different memory location to store the data. The address of this memory location is stored in the box created.

Example: let user = {name: ‘ram’ };

Let admin = user;

Admin.name = ‘ikram;

Console.log(user.name);

o/p: ‘ikram’

* when the value of admin is changed it automatically changes the value of user as well.

This happens because both ‘user’ and ‘admin’ are storing the address of the memory location. And when one changes the values in the allocated memory it is reflected in the other as well.

We can further elaborate it we can say that; copy by reference is like having two keys of the same room shared between ‘admin’ and ‘user’. I one of them alters the arrangement of the room the other would experience it adds well.

2.How to copy by value a composite datatype(array+objects).

Ans: Arrays, objects, functions are all of object type which comes under composite data types. As we know variable holds data in case of composite data type it holds reference that is address of that particular value in memory.

var a= 10; // here a holds the value 10.

var b= [10,20] // b holds some address like 8023 etc.. not 10 and 20.

Therefore we can’t clone data in composite data types. To do that spread operator is used, that is three dots (…), it spreads the elements of that particular array or object and its values can be used to assign to some other variable.

const readline = require('readline')

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

inp.on("close", () => {

var a = [1,2,3,4,5];

var b = […a];

var c = a;

a[0]=99;

console.log(a);

console.log(b);

console.log(c);

});

Output:[99, 2, 3,4,5]

[1,2,3,4,5]

[99,2,3,4,5]

Here , I used spread operator. Therefore by changing the value in array a didn’t affect the array c. Whereas ,I simply copied it that is I gave reference so, by changing a it also changes c.