Assignment 1:

1)

const pi = 3.14;

pi = 23; //error as const doesnt allow value change

2)

if(true){

let a =10;

}

console.log(a); //error as let: scope=block

3)

var Order = {

id: 1,

title: "coke",

price: 100,

getOrder(){

return this.id +" "+ this.title;

},

getPrice(){

return this.price;

}

};

let copiedObj = Object.assign({}, Order);

4)

class Object1{

name1;

length1;

constructor(name1:string){

this.name1 = name1;

this.length1 = name1.length;

}

}

let displayArray = (...objects: Array<object>) =>{

//console.log("objects array: ",objects);

for(let i in objects)

console.log(objects[i]);

}

let names = ['tom', 'ivan', 'Jerry'];

let [name1, name2, name3] = names;

const obj1 = new Object1(name1);

const obj2 = new Object1(name2);

const obj3 = new Object1(name3);

displayArray(obj1,obj2,obj3);

5)

//5.a question

var add = () => 21;

//5.b question

function userFriends(username: string, ...friends: Array<string>){

console.log("username: ",username);

console.log("friends: ",friends);

}

userFriends('Tony Stark','Natasha Romanoff','Thor','Steve Rogers','Hulk','Hawk Eye');

//5.c question

function capitals(...names: Array<string>){

console.log();

for(let i in names)

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

}

capitals('Tony Stark','Natasha Romanoff','Thor','Steve Rogers','Hulk','Hawk Eye');

6)

var laptop = "HP", model = "Pavillion", deskno = 123, client='Lewis Hamilton';

var ticket = `hi Sysnet, this is ${client} - deskNo.: ${deskno}, and my laptop: ${laptop},

model: ${model}, isn't working. Reques replacement.`;

console.log(ticket);

7)

let array = [1,2,3,4];

let [ele1, ele2, ele3, ele4] = array;

console.log("3rd element: ",ele3);

var organization = {

nameObj: "Capgemini",

address: { country: "France", pincode: 75017}

}

let {nameObj, address: a} = organization;

console.log("deep match destructured pincode: ",a.pincode);

8)

class Accounts{

id;

name;

balance;

constructor(id:number,name:string,balance:number)

{

this.id=id;

this.name=name;

this.balance=balance;

}

}

class SavingsAccount extends Accounts{

interest;

constructor(id:number,name:string,balance:number,interest:number)

{

super(id,name,balance);

this.interest=interest;

}

}

class CurrentAccount extends Accounts{

cash\_credit;

constructor(id:number,name:string,balance:number,cash\_credit:number)

{

super(id,name,balance);

this.cash\_credit=cash\_credit;

}

}

const saving1= new SavingsAccount(1,"abc",20000,2344);

const saving2= new SavingsAccount(2,"bbc",20000,2344);

const current1= new CurrentAccount(3,"abc",20000,2344);

const current2= new CurrentAccount(4,"abc",20000,2344);

console.log(saving1.balance + saving2.balance + current1.balance + current2.balance );