Es6 and typescript assignment

Assignment 1

Q1

function problem1()

{

const value = 5;

value = 10; // gives error

//Q2

function problem2()

{

if(6>2)

{

var b = 50;

}

console.log(b); //gives error

//Q3

let Order = {

id: 100,

title:'Pav Bhaji',

price: 250,

printOrder()

{

console.log(this.id);

},

getPrice()

{

console.log(this.price);

}

}

const copyobj = Object.assign({},Order);

console.log(copyobj);

//Q4

let array= ['hello', 'this', 'is','arraystring','to','object'];

let newArr = array.map(item => {

return {

'name': item,

'length':item.length

}

})

console.log(newArr);\*/

//Q5 A

let add = function (value =10)

{

return value;

}

console.log("The default values is : " + add());

//Q5 B

let userFriends = function(username , ...userfriends)

{

console.log( "username:" username);

for(let i in userfriends)

console.log(userfriends[i]);

};

let username = "Sushmita";

let userfriends = ["Riya", "Priya", "Chia"];

userFriends(username, userfriends);

//Q5 C

let printCapitalNames = function(...names)

{

for(let i in names)

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

}

let nameArray = ["Shreya", "Shriya", "Pakhi", "Priya", "Mia"];

printCapitalNames(...nameArray);

//Q6

let myname = “Pooja";

let laptopmodel = 'HP 234';

let deskno = 10001;

console.log(`Dear Sysnet, I am facing an issue with my pc I kindly request you to look into it.

My details are as follows: Name : ${name}, LaptopModel: ${laptopmodel}, Deskno: ${deskno} `);

//Q7A

let food = ['Pav Bhaji', 'Masala Dosa','Palak Paneer','PastaBolognese'];

let [item1, item2, item3, item4] = food;

console.log(item3);

//Q7B

let org ={name:'thor',address:{city:'asgard',pincode:12345}};

let {name,address.city,address.pincode}= org;

console.log(address.pincode);

//Q8

class Account{

constructor(id,name,balance)

{

this.id = id;

this.name = name;

this.balance = balance;

}

}

class SavingsAccount extends Account{

constructor(id,name,balance,interest)

{

super(id,name,balance);

this.interest = interest;

}

totalBalance = () => { this.balance += this.interest;

console.log(this.balance);};

}

class CurrentAccount extends Account{

constructor(id,name,balance,cashCredit)

{

super(id,name,balance);

this.cashCredit = cashCredit;

}

totalBalance = () => { this.balance += this.cashCredit;

console.log(this.balance);};

}

let S1 = new SavingsAccount(1001,'Sushmita',40000,4000);

let S2 = new CurrentAccount(10003, 'Riya', 30000,3800);

S1.totalBalance();

S2.totalBalance();\*/

Assignment 2

1//

class name

{

function nextFibonacci(curent\_value)

{

let next\_value=current\_value \*(1+ Math.sqrt(5))/2.0;

return Math.round(next\_value);

}

let current\_value=5;

console.log("next value is:"+nextFibonacci(current\_value));

}

//2

let armstrong = {

[Symbol.iterator](){

let low=10,value;

return{

next(){

function armstrong(low) {

var num = low+1;

while(true){

var str = num.toString();

var sum = 0;

for(var i = 0;i<str.length;i++){

sum = sum + Math.pow(parseInt(str[i]),str.length);

}

if(sum == num)

return sum;

else

num++;

}

}

[value,low]=[armstrong(low),armstrong(low)];

return{value};

}

};

};

};

for (n of armstrong){

if(n>10000){

break;

}

console.log(n);

}

function \* armstrong11(low) {

//var num = low+1;

while(true){

if(low>1000){

return{done:true,'value':low};

}

//console.log(low);

low++;

do{

var str = low.toString();

var sum = 0;

for(var i = 0;i<str.length;i++){

sum = sum + Math.pow(parseInt(str[i]),str.length);

}

if(sum == low){

//[value,low]=[sum,sum];

//var value = sum;

low = sum;

break;

}

else

low++;

}while(true);

yield low;

}

}

var arr = armstrong11(10);

let temp = arr.next();

console.log(temp);

temp = arr.next();

console.log(temp);

temp = arr.next();

console.log(temp);

temp = arr.next();

console.log(temp);

temp = arr.next();

console.log(temp);

temp = arr.next();

console.log(temp);

\\q4

let chatroom1 = new Map();

let chatroom2 = new Map();

let user1 = new Set(["hii"]);

let user2 = new Set(["hello"]);

let user3 = new Set(["hello world"]);

let user4 = new Set(["c2u1","Bilanshi nagin nighali"]);

let user5 = new Set(["c2u2","Abhi maja aayaga na bhidu"]);

let user6 = new Set(["c2u3","Kya gunda banega re tu"]);

chatroom1.set("user1 Raju",user1);

chatroom1.set("user2 Shyam",user2);

chatroom1.set("user3 Baburao",user3);

chatroom2.set("user1 Raju",user4);

chatroom2.set("user2 Farhan",user5);

chatroom2.set("user3 Rancho",user6);

function printusersofchat(chatroom){

//console.log("\nusers form chatroom :")

for(let key of chatroom.keys()){

console.log("\n",key);

}

}

function printchatsofchat(chatroom){

let chats = new Set();

for(let key of chatroom.values()){

// console.log("\n",key);

//chats.add(key);

key.forEach(chats.add,chats);

}

console.log("\n",chats);

}

console.log("\nUsers of chatroom 1 :");

printusersofchat(chatroom1);

console.log("\nUsers of chatroom 2 :");

printusersofchat(chatroom2);

console.log("\n Chats from chatroom 1 :");

printchatsofchat(chatroom1);

console.log("\n Chats from chatroom 2 :");

printchatsofchat(chatroom2);

Assignment 3

//Q1

var x:number , y:number;

var pro1 = new Promise(function(resolve,reject)

{

if(1>0)

resolve(x=15);

else

reject(4);

});

var pro2 = new Promise(function(resolve,reject)

{

if(1>0)

resolve(y =10);

else

reject(8);

});

Promise.all([pro1, pro2]).then(values =>{console.log(x + y);});

//Q2

class Account{

constructor(id,name,balance)

{

this.id = id;

this.name = name;

this.balance = balance;

}

}

class SavingsAccount extends Account{

constructor(id,name,balance,interest)

{

super(id,name,balance);

this.interest = interest;

}

totalBalance = () => { this.balance += this.interest;

console.log(this.balance);};

}

class CurrentAccount extends Account{

constructor(id,name,balance,cashCredit)

{

super(id,name,balance);

this.cashCredit = cashCredit;

}

totalBalance = () => { this.balance += this.cashCredit;

console.log(this.balance);};

}

let S1 = new SavingsAccount(1001,'Shreya',40000,4000);

let S2 = new CurrentAccount(10003, 'Supriya', 30000,3800);

S1.totalBalance();

S2.totalBalance();\*/

//Q3

/\*interface Printable{

fname?:string;

id?: number;

area?:Number;

};

let circle :Printable ={

area: 720,

};

let employee :Printable = {

fname: "Pooja",

id:10001,

};

function printAll(circle, employee){

console.log(circle);

console.log(employee);

}

printAll(circle,employee);