JS Exam

1. You have a list of users given to you by the backend. You’ll need to filter out the users that have paid for their subscription vs those that haven’t paid into their own arrays. Additionally, you’ll need to filter out all users that are employees into their own array. Complete this by using “for” loops and then again using Array.prototype.filter

const users = [

{name: ‘John’, age: 24, subscription: ‘paid’, employee: false},

{name: ‘Samantha’, age: 4, subscription: ‘paid’, employee: false},

{name: ‘Jim’, age: 21, subscription: ‘unpaid’, employee: true},

{name: ‘Brooke’, age: 21, subscription: ‘paid’, employee: false},

{name: ‘Tom’, age: 21, subscription: ‘unpaid’, employee: false},

{name: ‘Katie’, age: 21, subscription: ‘unpaid’, employee: true}

]

1. List the base Data Types of Javascript and explain their use:
2. Describe the use and purpose of a function in code.
3. What will be the output from the following code?

function\* employeeBounuses(emps) {

for(let i = 0; i < emps.length; i++) {

if (emps[i].yearsEmployeed > 10) {

yield emps[i].salary \* .3

} else if (emps[i].yearsEmployeed > 5) {

yield emps[i].salary \* .2

} else {

yield emps[i].salary \* .1

}

}

}

function main() {

let employees = [{

name: 'Bob',

yearsEmployeed: 6,

salary: 80000

}, {

name: 'Jane',

yearsEmployeed: 3,

salary: 56000

}, {

name: 'Tim',

yearsEmployeed: 11,

salary: 110000

}]

const bonusCalculator = employeeBounuses(employees)

console.log(`Pay ${employees[0].name} ${bonusCalculator.next().value} for his bonus`)

console.log(`Pay ${employees[1].name} ${bonusCalculator.next().value} for her bonus`)

console.log(`Pay ${employees[2].name} ${bonusCalculator.next().value} for his bonus`)}

main ()

1. How would you access a key’s value on an object? If the user selected what data to see, how would you access the key’s value?
2. The following code is stubbed out for a callback, complete it:

let employees = [{

name: 'Bob',

id: 'e582a642-e400-453f-b0f1-4aaa0a419cb5',

yearsEmployeed: 6,

salary: 80000

}, {

name: 'Jane',

id: '239ae3ce-da74-46cd-bd98-b98e9406a80b',

yearsEmployeed: 3,

salary: 56000

}, {

name: 'Tim',

id: '7e6a2835-0389-42ca-a4f6-321386fd6947',

yearsEmployeed: 11,

salary: 110000

}]

function fakeBackend(userId, callback) {

//Check to see if the userId exists for one of the employees

//if the userId doesn't exist call the callback with an error object

//for the first parameter with the fields err set to true, and msg set

//to 'Unkown employee.' Otherwise return the employee info

}

function main() {

const callback = ( ) => {

//If there's an error call fakeBackend with a known employee id

//When the callback gets a successful call, then console.log out the employee info

}

fakeBackend('some-unknown-user-handle', callback)

}

main ()

1. Alter the above problem to use a promise instead of a callback