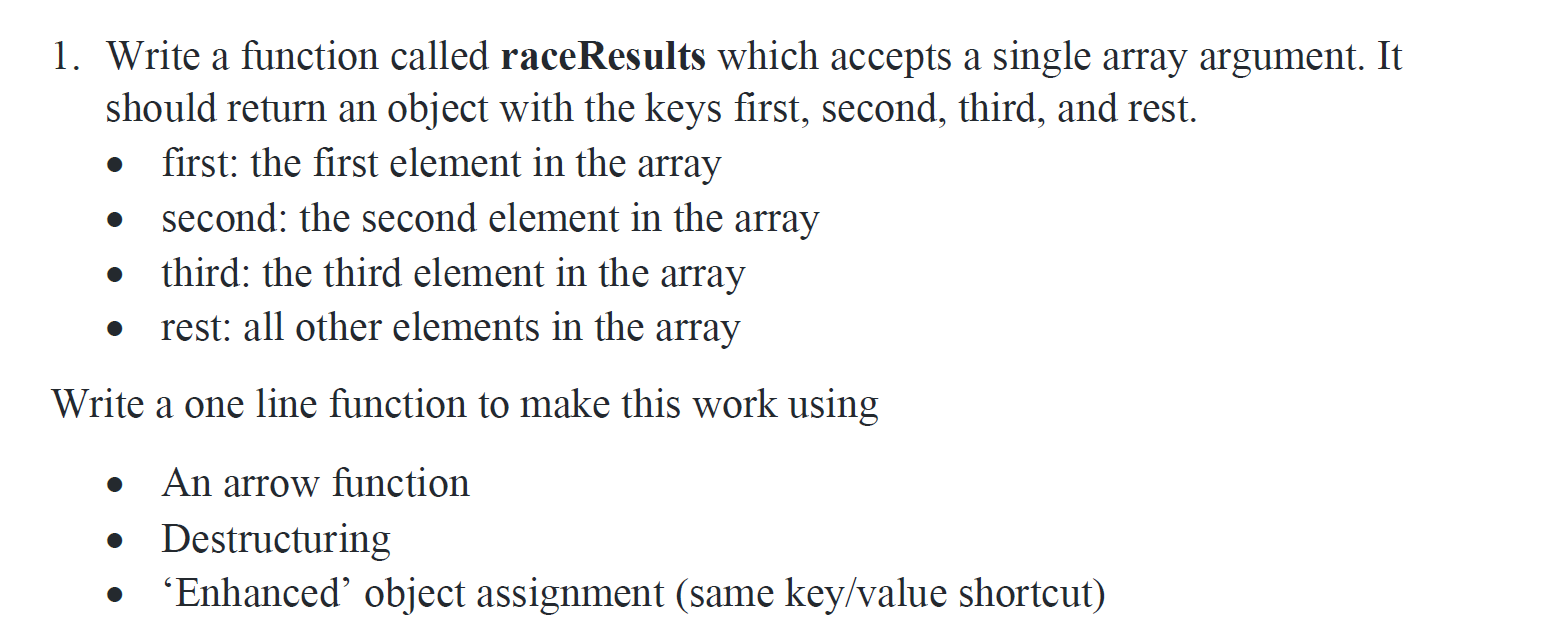
**Lab – 08**

Name: Ammar Ahmed

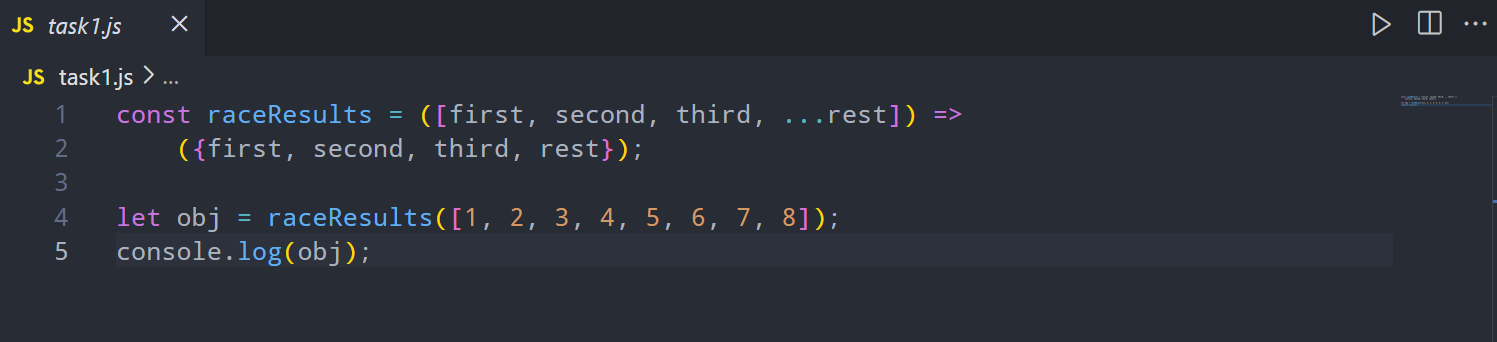
CMS: 023-19-0107

Sec: ‘A’

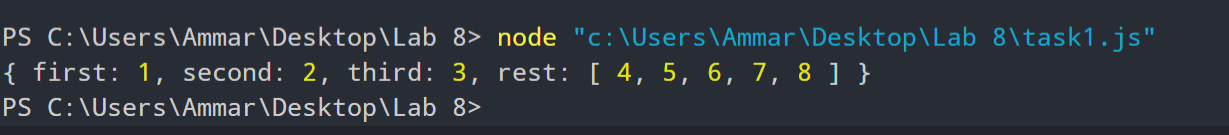
Web Engineering

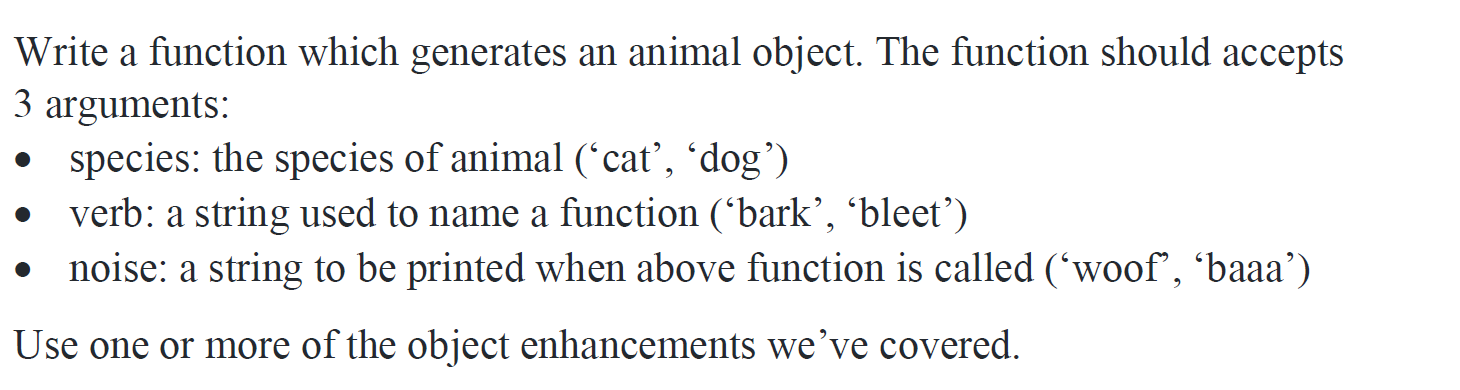
**1.** 

**Code:**

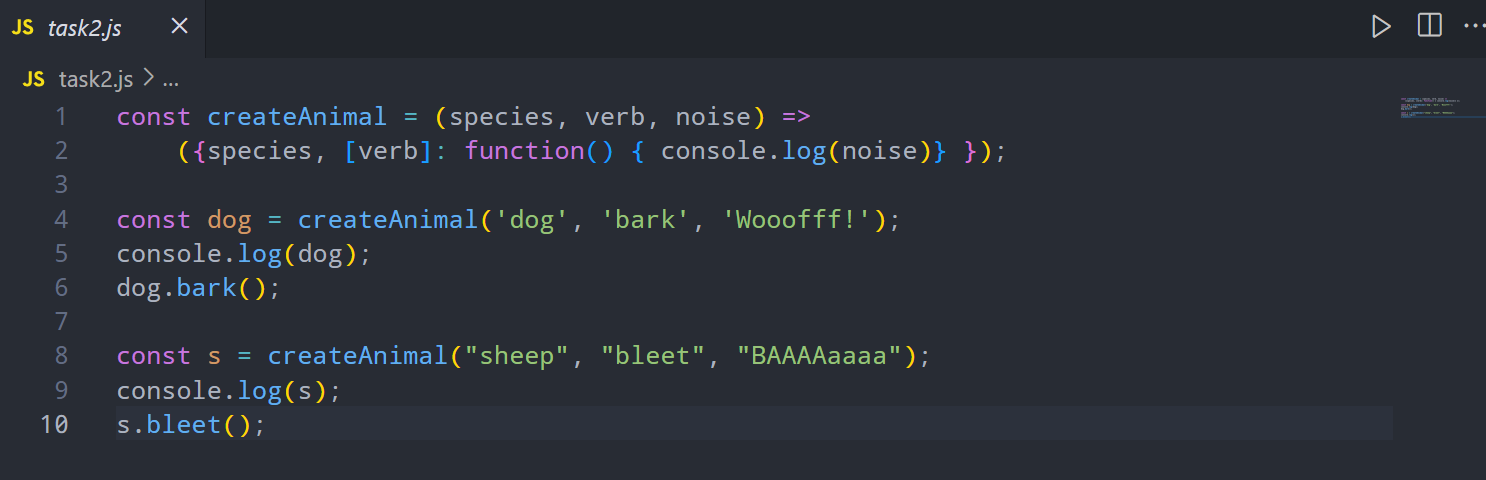
****

**Output:**

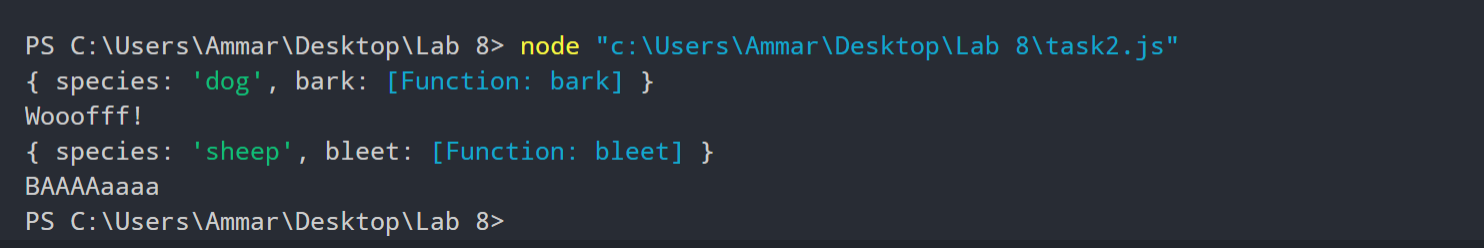
****

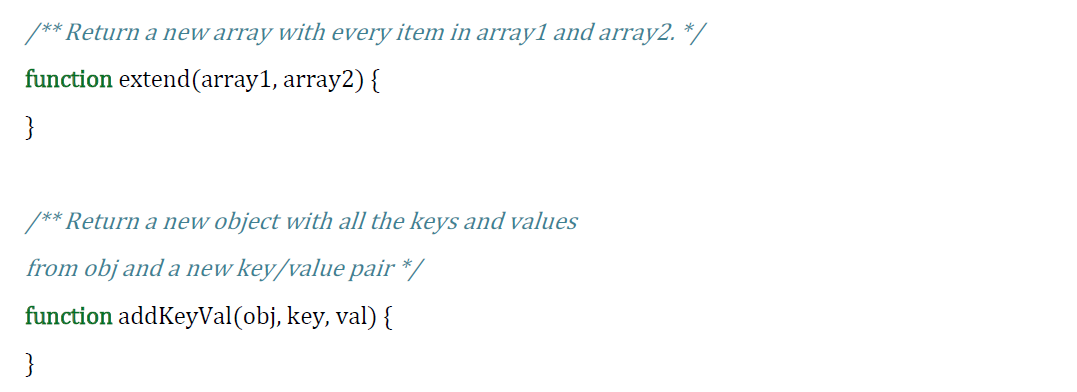
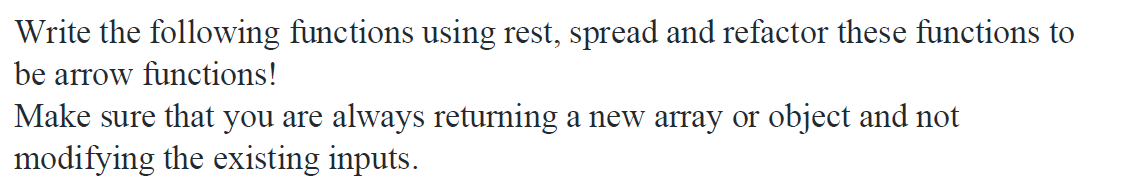
**2.** 

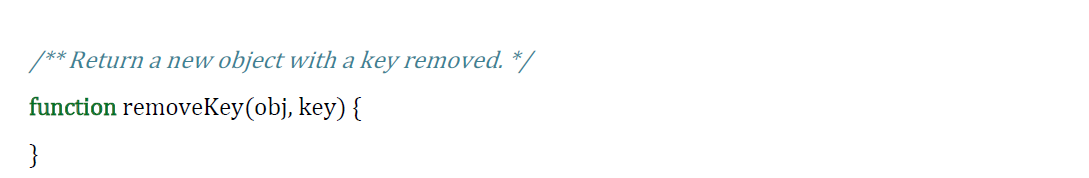
**Code:**

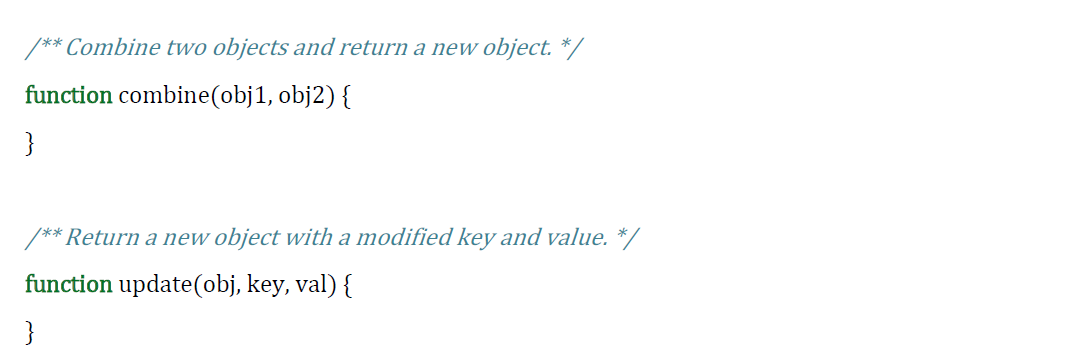
****

**Output:**

****

**3.** 

****

****

**Code:**

*/\*\* Return a new array with every item in array1 and array2. \*/*

const extend = (arr1, arr2) => [...arr1, ...arr2];

console.log(extend([1, 2, 3, 4], [9, 0]));

*/\*\* Return a new object with all the keys and values from obj*

*\* and a new key/value pair \*/*

const addKeyVal = (obj, key, val) => ({...obj, [key]:val});

const obj = {name: 'Ammar', age: '18', isStd: 'true'};

console.log(addKeyVal(obj, 'sem', 'Six'));

*/\*\* Return a new object with a key removed. \*/*

const removeKey = (obj, key) => {

    let obj2 = {...obj};

    delete obj2[key];

    return obj2;

}

console.log(obj);

console.log(removeKey(obj, 'isStd'));

*/\*\* Combine two objects and return a new object. \*/*

const combine = (obj1, obj2) => ({...obj1, ...obj2});

const obj2 = {

    fieldOfStudy: 'Computer Science',

    interests: ['Machine Learning/AI', 'Literature', 'Astronomy', 'Behavioral Biology']

};

console.log(combine(obj, obj2));

*/\*\* Return a new object with a modified key and value. \*/*

const update = (obj, key, val) => {

    let obj2 = {...obj};

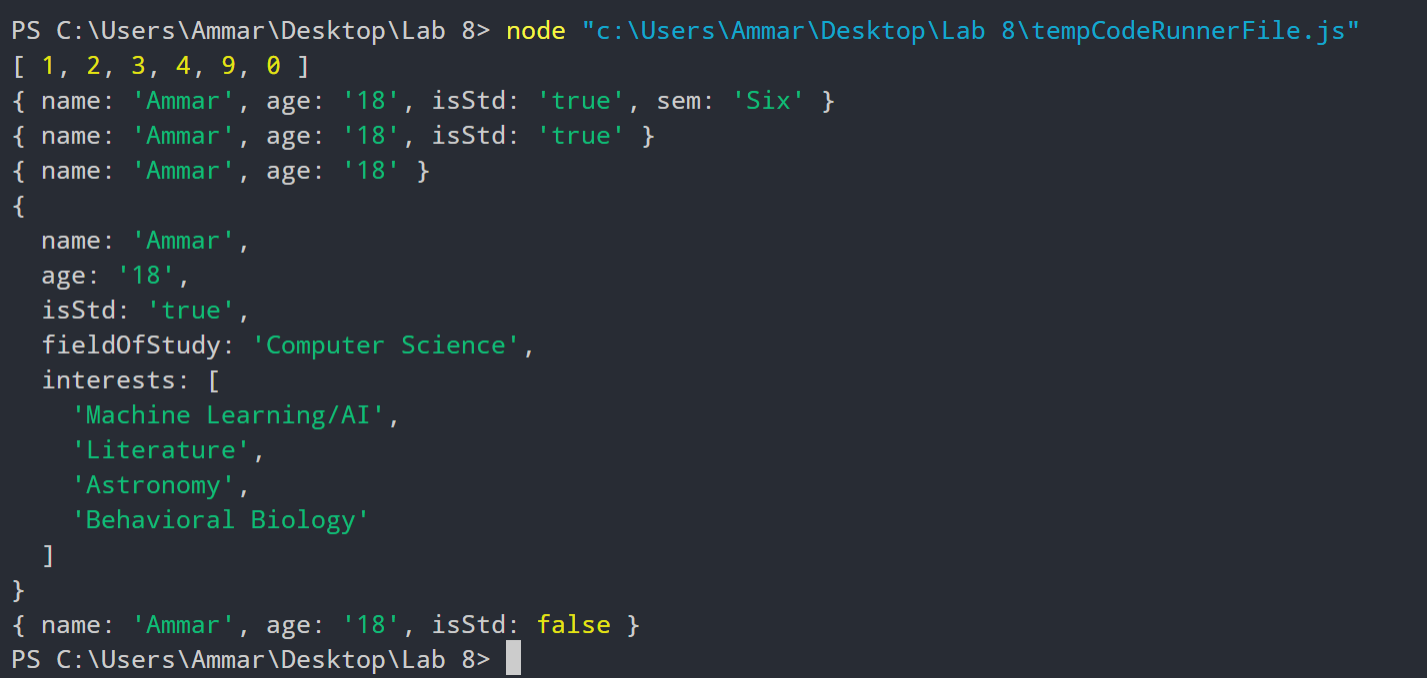
    obj2[key] = val;

    return obj2;

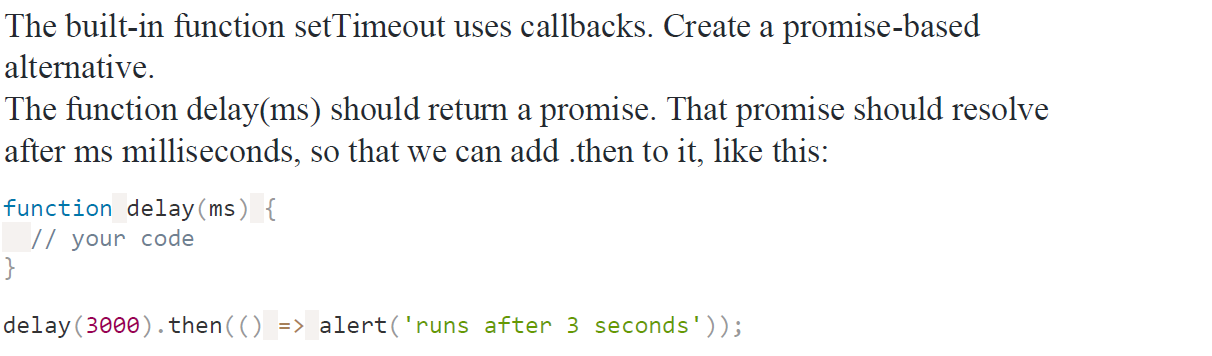
}

console.log(update(obj, 'isStd', false));

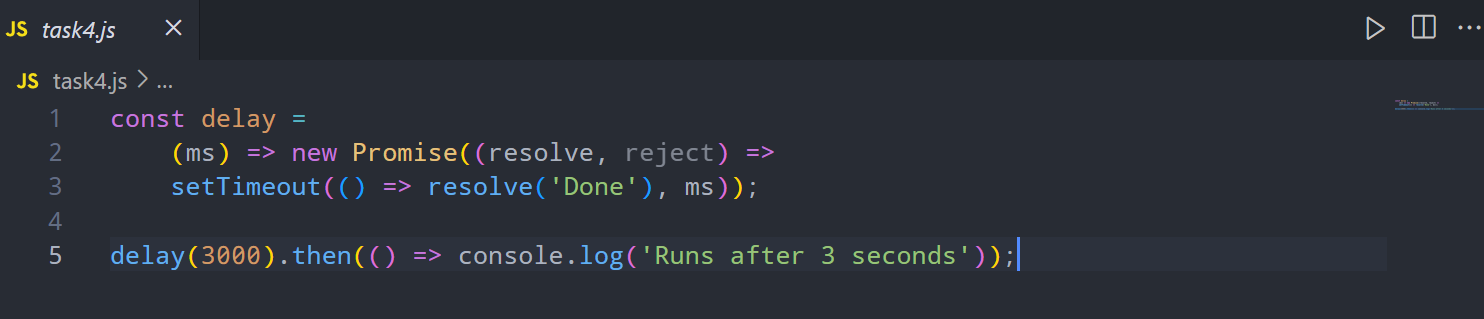
**Output:**

****

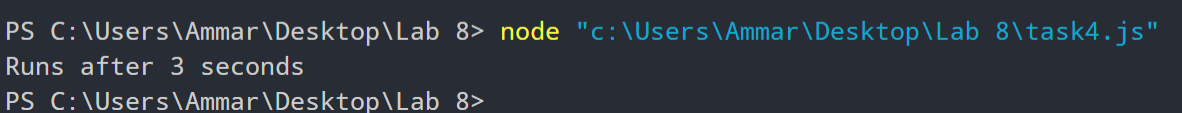
**4.**



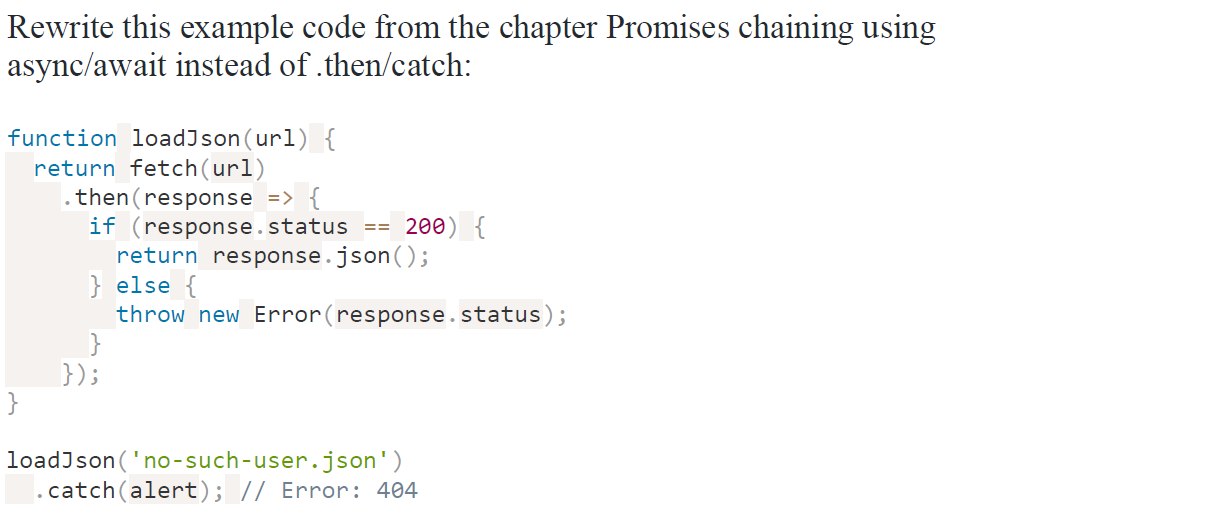
**Code:**

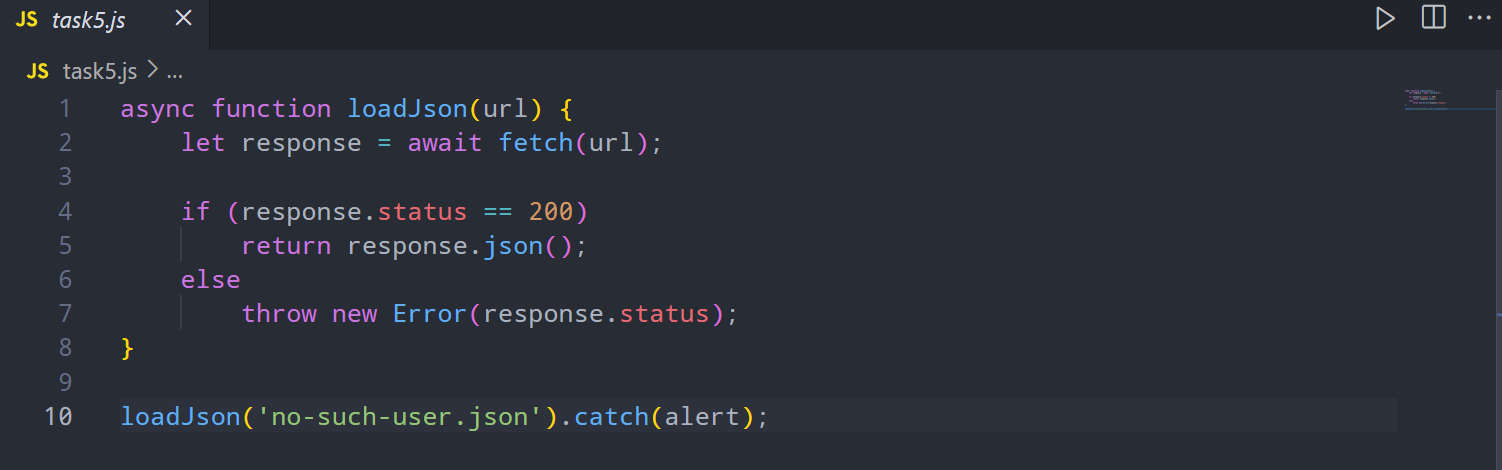
****

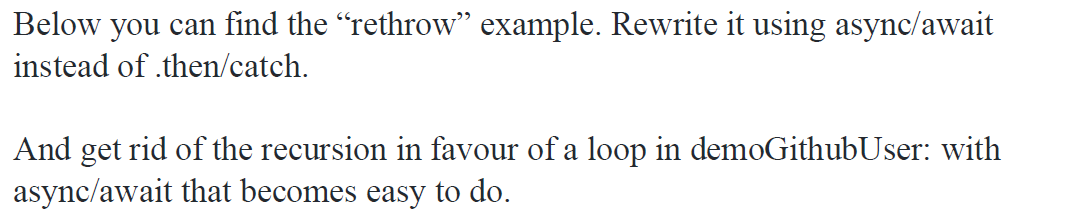
**Output:**

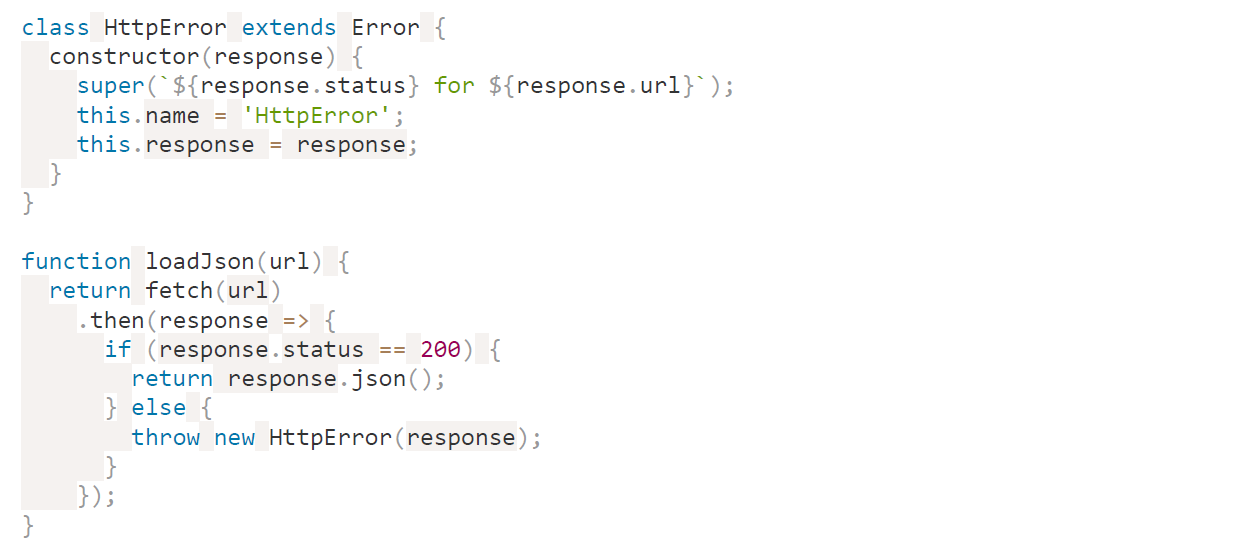
****

**5.**

****

**Code:**

**6.** ****

****

****

**Code:**

class HttpError extends Error {

    constructor(response) {

        super(

            `${response.status} for ${response.url}`

        );

        this.name = 'HttpError';

        this.response = response;

    }

}

async function loadJson(url) {

    let response = await fetch(url);

    if (response.status == 200)

        return response.json();

    else

        throw new HttpError(response);

}

async function demoGithubUser() {

    let user = '';

    while (!user) {

        let name = prompt("Enter a name?", "iliakan");

        try {

            user = await loadJson(`https://api.github.com/users/${name}`);

        } catch(err) {

            if (err instanceof HttpError && err.response.status == 404) {

                alert("No such user, please reenter.");

                continue;

            } else {

                throw err;

            }

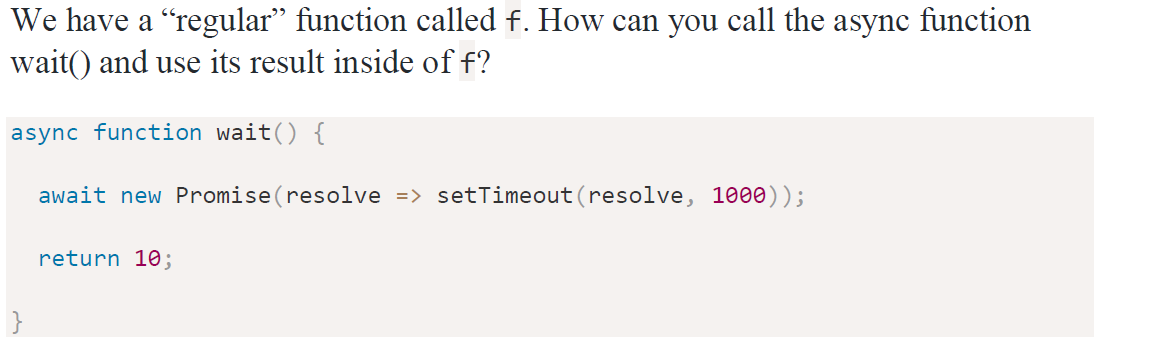
        }

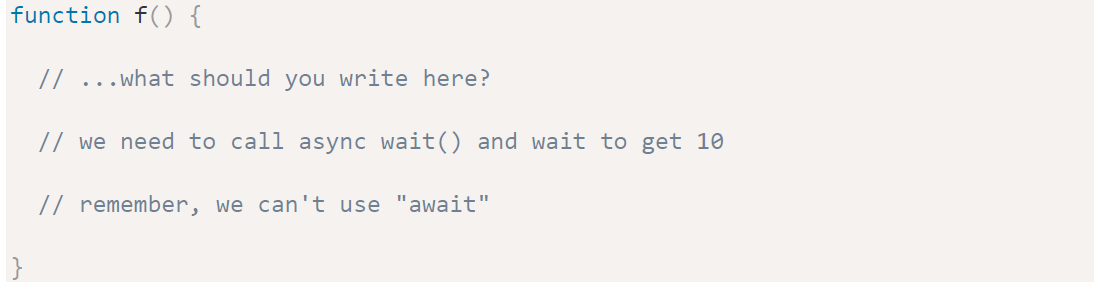
    }

    alert(`Full name: ${user.name}.`);

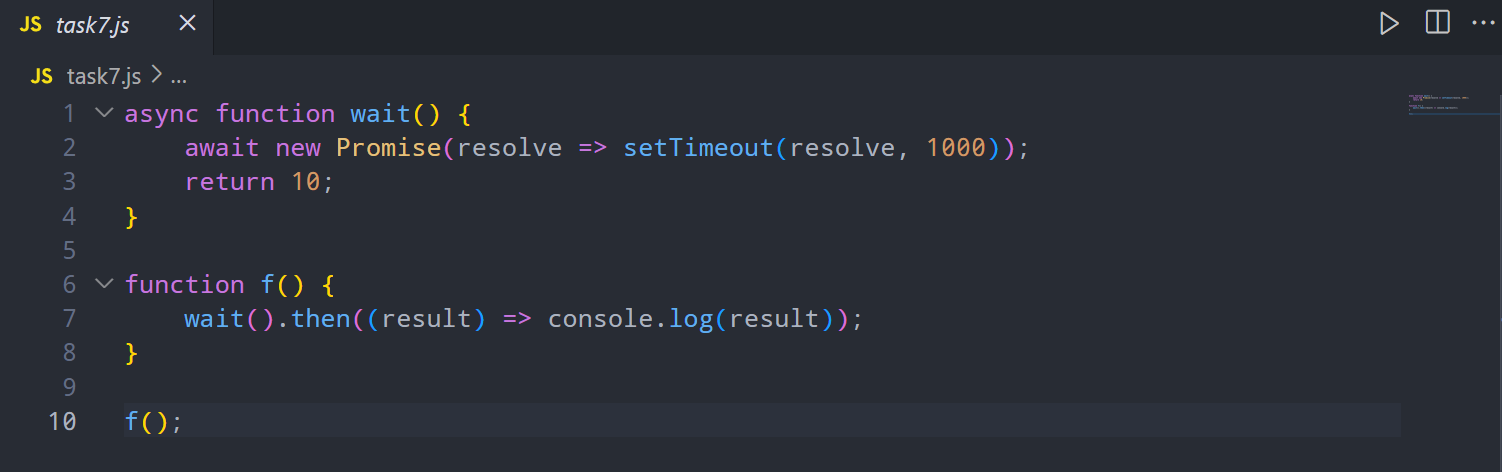
    return user;

}

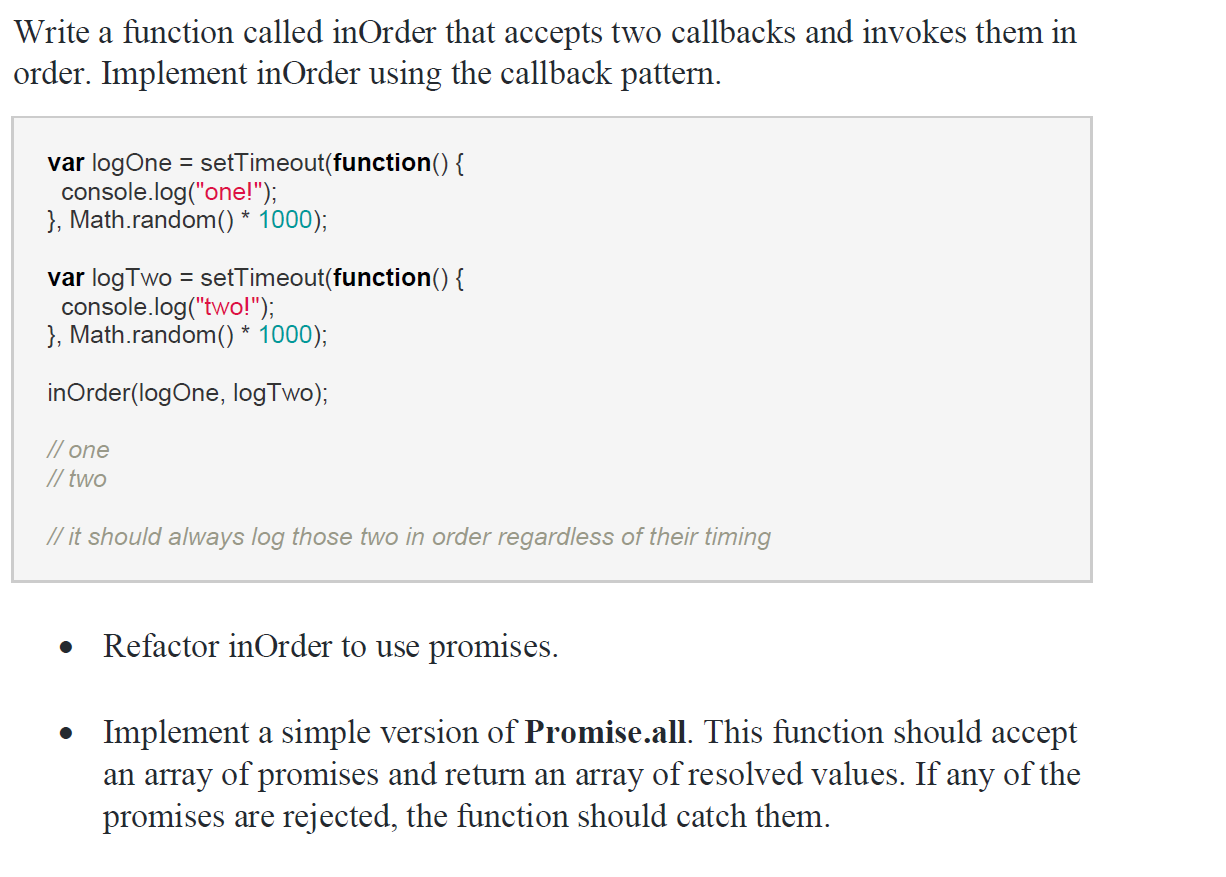
**7.**

****

**Code:**

****

**8.**



**Code:**

****

**END**