



College of Engineering, Construction & Living Sciences  
Bachelor of Information Technology  
ID607001: Introductory Application Development Concepts  
Level 6, Credits 15  
**In-Class Activity: ES6 Basics 2**

## Instructions

The purpose of this in-class activity is to familiarise yourself with more complex constructs such as **map**, **filter** & **reduce**. Also, you will look at how to read & process data from a local file. The following eight questions will require a little more thought than the previous in-class activity.

## Submission

You must submit all program files via **GitHub Classroom**. Here is the URL to the repository you will use for your code review – [https://classroom.github.com/a/\\_6KSahyX](https://classroom.github.com/a/_6KSahyX). If you wish to have your code reviewed, message the course lecturer on **Microsoft Teams**.

## Getting Started

Open your repository in **Visual Studio Code**. Create a new file called **02-in-class-activity.js**. In **02-in-class-activity.js**, add the following:

```
console.log('Hello, World!')
```

Open a **terminal** & run the following command:

```
node 02-in-class-activity.js
```

If the output is **Hello, World!**, then you are ready to start coding.

## Problem 1:

For each element in **nums**, calculate its power of two & return as an **array** using the **map** function.

```
const nums = [2, 4, 6, 8, 10]

const powOfTwo = // Write your solution here
console.log(powOfTwo)

// Expected output:
// [4, 16, 36, 64, 100]
```

### Problem 2:

For each element in **temps**, convert its value from fahrenheit to celsius & return as an **array** using the **map** function. Round each value to the nearest two decimal places using the **toFixed** function.

```
const temps = [65, 45, 25, 5]

const fahToCel = // Write your solution here
console.log(fahToCel)

// Expected output:
// [18.33, 7.22, -3.89, -15.00]
```

### Problem 3:

Using the **filter** function, return countries that have a population of less than 1000000000 (one billion).

```
const countries = [
  { name: 'Brazil', population: 213445417 },
  { name: 'China', population: 1339330514 },
  { name: 'India', population: 1352642280 },
  { name: 'Russia', population: 142320790 },
  { name: 'United States of America', population: 332475723 }
]

const countriesWithPopLessThanOneBil = // Write your solution here
console.log(countriesWithPopLessThanOneBil)

// Expected output:
// [
//   { name: 'Brazil', population: 213445417 },
//   { name: 'Russia', population: 142320790 },
//   { name: 'United States of America', population: 332475723 }
// ]
```

### Problem 4:

Using the **filter** function, return animals that are native to New Zealand.

```
const animals = [
  { name: "Cassowary", native_country: "Australia" },
  { name: "Kiwi", native_country: "New Zealand" },
  { name: "Little Blue Penguin", native_country: "New Zealand" },
  { name: "Bald Eagle", native_country: "United States of America" }
]

const nativeAnimals = // Write your solution here
console.log(nativeAnimals)
```

```
// Expected output:
// [
//   { name: 'Kiwi', native_country: 'New Zealand' },
//   { name: 'Little Blue Penguin', native_country: 'New Zealand' }
// ]
```

### Problem 5:

Using the **reduce** function, return the total **price** for the given **groceries** array of **objects**.

```
const groceries = [
  { name: 'Chicken', price: 10 },
  { name: 'Butter', price: 5 },
  { name: 'Lettuce', price: 2 },
  { name: 'Steak', price: 20 },
]

const groceriesTotal = // Write your solution here
console.log(groceriesTotal)

// Expected output:
// 37
```

### Problem 6:

Using the **reduce** function, return an **object** where the **key** is the name of the ice cream flavour, i.e., chocolate & the **value** is an **integer** that represents the total count for that flavour, i.e., 3.

```
const iceCreamFlavours = [
  'vanilla', 'chocolate', 'strawberry',
  'vanilla', 'mango', 'vanilla',
  'chocolate', 'strawberry', 'mango',
  'orange', 'chocolate'
]

const iceCreamFlavourCount = // Write your solution here
console.log(iceCreamFlavourCount)

// Expected output:
// { vanilla: 3, chocolate: 3, strawberry: 2, mango: 2, orange: 1 }
```

### Problem 7:

Using the **readFile** function, read **nursery-rhyme.txt** located in the **formative assessments** directory. For each word in **nursery-rhyme.txt**, convert it to **lowercase** using the **map** function.

```
// Expected output:
// [
//   'old',      'macdonald', 'had',
//   'a',        'farm,',    'e-i-e-i-o!',
//   'and',      'on',      'his',
//   'farm',     'he',      'had',
//   'a',        'cow,',    'e-i-e-i-o!',
//   'with',     'a',      'moo-moo',
//   'here',     'and',     'a',
// ]
```

```
//      'moo-moo',    'there,',    'here',  
//      'a',          'moo,',    'there',  
//      'a',          'moo,',    'everywhere',  
//      'a',          'moo-moo,',  'old',  
//      'macdonald',  'had',    'a',  
//      'farm,',      'e-i-e-i-o!'  
// ]
```

### Problem 8:

Using the **readFile** function, read **users.json** located in the **formative assessments** directory. Using the **filter** function, return **users** who are **Senior Lecturers**.

```
// [  
//   {  
//     first_name: 'Faisal',  
//     last_name: 'Hassan',  
//     position: 'Senior Lecturer'  
//   },  
//   {  
//     first_name: 'Joy',  
//     last_name: 'Gasson',  
//     position: 'Senior Lecturer'  
//   }  
// ]
```