The following exercises are related to the use of JavaScript.

1. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.

Solution:

```
//Make a folder for your work called lab1
// Open a command prompt in that folder by typing cmd at the top of file explorer.
-------------------------------------------------------------
//Alternatively you could navigate to the folder using window command prompt
cd lab1
-------------------------------------------------------------
//Add a file to your repository preferably a README.md

-------------------------------------------------------------
git init
Initialized empty Git repository in /Users/~/lab1/.git/
-------------------------------------------------------------
git add .
Add all files to the staging area. They will not be saved at this point
-------------------------------------------------------------
git commit -m "Empty first commit."
[master (root-commit) 6583cf6] Empty first commit.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
-------------------------------------------------------------

git remote add origin https://github.com/myusername/WHATEVER.git
You will need to add the path to your GitHub Repository here
-------------------------------------------------------------
git push -u origin main
Add you file to GitHub (remote repository)
-------------------------------------------------------------
```

2. In this exercise you are required to create a class in JavaScript called BMI, that will take height in meters and weight in Kg as arguments and will calculate that persons Body Mass Index (BMI). To complete this exercise you will need to do the following:
   (a) Create a new file in VS Code called bmi.js
   (b) Create a class called BMI
      i. Add a constructor to the class.
      ii. The constructor should take two arguments (height & weight).

iii. The class should contain a method called calculateBMI.

iv. This method will return the individuals bmi using the formula:
```
let bmi = this.weight/(this.height**2);
```

(c) Create an instance of the class and invoke the calculateBMI method.

Solution:

```
class BMI {
    constructor(height, weight){
        this.height = height;
        this.weight = weight;
    }

    calculateBMI(){
        let answer = this.weight/(this.height**2);
        return answer;
    }
}

let myBMI = new BMI(2, 90);
let bmi = myBMI.calculateBMI();
console.log(bmi);
```

3. In this exercise, the aim is to implement a class called vehicle and a child class called Car. The Vehicle class should take make, model and year as arguments. It also should include a method that will write this information (make, model, year) to the console. The Car class will inherit from Vehicle class and also have a method that writes Number of Doors to the console.

(a) Create a new file in VS Code called cars.js

(b) Add a new Vehicle class.

i. The vehicle class should contain a constructor.

ii. The constructor should contain three arguments (make, model, year)

iii. Add a method to this class called Information that logs make model and year to the console.

iv. create an instance of this class and ensure that it is working properly.

(c) Add a new class called Cars that inherits from the Vehicle class.

i. The cars class should contain a constructor.

ii. The constructor should contain four arguments (make, model, year, doors)

iii. The constructor should invoke the parents class constructor and pass it three argument s for make model and year.

iv. The car class should contain a method called info Information.

v. This method will invoke the Information method on the parent class and also wirite the number of doors to the console.

(d) create an instance of this class and ensure that it is working properly.

Solution:

```
class Vehicle {
    constructor(make, year, model) {
        this.make = make;
        this.model = model;
        this.year = year;
    }

    Information() {
        console.log(`Make: ${this.make}.`);
        console.log(`Model: ${this.model}.`);
        console.log(`Year: ${this.year}.`);
    }
}

class Car extends Vehicle {
    constructor(make, model, year, doors) {
        super(make, year, model);
        this.doors = doors;
    }

    Information() {
        super.Information();
        console.log(`Doors: ${this.doors}`);
    }
}

let myCar = new Car('VW', 22, 'Golf', 4);
myCar.Information();
```