IMY 220
Practical 4:
JavaScript – ES6

Due: Friday 24 August @ 11:30

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

You will get 0 if you do not submit on ClickUP.

Instructions

- For this practical, you will need to update last week's assignment to use ES6 where appropriate and extend the *SynthHandler* class. Download *index.html* and *script.js* from ClickUP, which contains the requirements for the *SynthHandler* class for Assignment 6. You **must** put your name and surname in the "author" meta tag.
- All of your code for this practical must be written inside script.js.

Section 1: Update to ES6

Update all the JavaScript in *script.js* to use appropriate ES6 counterparts. Redefine *SynthHandler* as an ES6 class and define variables in the most appropriate way. Also redefine all anonymous functions as arrow functions and respectively use block bodies or expression bodies where appropriate. Use the destructuring assignment to assign variables and use template strings where appropriate.

Section 2: Extend SynthHandler class

Extend the *SynthHandler* class to include a *getter*, *setter*, and a new function.

- The *getter* should work in such a way that if you define a *SynthHandler* instance and access its member variable, for example, *handler.synths* (where handler is a SynthHandler instance), it should return an array of summaries in the same format as the *getSummaries* function.

- The *setter* should work in such a way that it sets the member variable to the given variable, for example *handler.synths = newSynths*. However, it should only set the member variable this way if the new variable is of the same format as the given *synths* and *drumMachines* variables, in other words, if it is an array of objects that each have a *name*, *manufacturer*, and *released* property. If the given variable does not meet those requirements, the member variable of the class must remain unchanged.

  You must also do this check without using any loops (including the *forEach* function) and you must use a *try-catch* statement which logs an appropriate error if the give variable is not of the correct format. (Use the JS *hasOwnProperty* function to check for properties.)

- Add a new function to *SynthHandler* called *getNamesList*, which returns an array of all the names of the synths in the object’s member variable. This must also be done without loops and must use appropriate ES6 syntax including the destructuring assignment to access the names.

Submit ONLY the following file(s) according to the submission instructions.

- *index.html*
- *script.js*