JavaScript Practice 002
Module 1.2

1.

Instruction:

- Write class **Book**. Then, add a method to the object that outputs information about the book.

Class Book

Constructor

- **Constructor(title, author, yearPublished)**: This initializes the properties 'title', 'author', and 'yearPublished'. If they are empty, it assigns **'no title', 'no author', and 0** respectively.

Properties

- **title**: type String
- **author**: type String
- **yearPublished**: type number

Methods

- **getInfo()**: This method return a string that combines the book's title, author, and year of publication.

Initiate code:

```
// Creating class Book
// Insert your code here
```

2.

Instruction:

- Create an object Book reference ‘practice 1’. Use a constructor **Constructor(title, author, yearPublished**. Additionally, use a 'getInfo()' method.

Initiate code:

```
// Insert your code here
```

3.

Instruction:

- Creating a library system using class **Book** and **Library**.

Class Book:

Constructor

- **Constructor(title, author, pages, genre):** This initializes the properties 'title', 'author' , ' pages' and 'genre'.

Properties

- **title**: type string
- **author**: type string
- **pages**: type number
- **genre**: type string

Methods

- **getSummary()** - Returns a **string summary** of the book ex: . "Title: Harry Wick, Author: Conan, Pages: 345, Genre: Sci-Fi"

Class Library:

Constructor

- **Constructor(name):** This initializes the properties 'name' and creates an empty array in the property **books**

Properties

- **name**: type string
- **books**: type array of **Book** objects, initially empty.

Methods

- **addBook(newBook)** - Accepts a **Book** object and adds it to the **books** array.
- **getBooksByGenre(genre)** - Returns an array of books that match the specified genre.
- **getTotalPagesByGenre(genre)** - Returns the total number of pages for all books of a specified genre.

4

Instruction:

- Create a 'Book' object and a 'Library' object for 'Practice 3'. Use a constructor and methods in both classes.

Initiate code:

```
// Insert your code here
```

5.

Instruction:

- Create class **Elevator** with the following specifications:

Class Elevator:

Constructor

- **Constructor(maxFloor, minFloor):** This initializes the properties 'maxFloor', 'minFloor' and set 'currentFloor' to 0

Properties

- **currentFloor**: The floor where the elevator currently is. , type number
- **maxFloor**: The highest floor the elevator can go to. , type number
- **minFloor**: The lowest floor the elevator can go to , type number

Methods

- **goUp()**: Increases **currentFloor** by 1, but not above **maxFloor**.
- **goDown()**: Decreases **currentFloor** by 1, but not below **minFloor**.
- **goToFloor(floor)**: Takes a floor number and sets **currentFloor** to that floor if it's within range.
- **displayFloor()**: Prints the current floor to the console.

6

Instruction:

- Create a 'Elevator' object for 'Practice 5'. Use a constructor and methods in class.

Initiate code:

// Insert your code here