Advanced JavaScript
Lab 2

Note: Make your own interface for the following tasks.

1. Create your own object that contains a list of numerical sequence, with the following details

- **Your constructor takes 3 parameters to define start, end of list and step**
- **The list should be private and filled with private method**
- **You can create getter and setter for the list if needed**
- **Allow the user to apply the following functionality to his created sequence**
  - **Append or prepend a new value**
  - **Dequeue or pop a value,**
  - **you have to ensure that you are pushing sequential value otherwise through exception**
  - **you have to ensure that there is no duplicated value otherwise through exception**
  - **override toString function to display you list**
- **you can add any property you need.**

2. create your box object that contains books objects, ensure that you can

- **count # of books inside box**
- **delete any of these books in box according to book title.**
- **create book object and add it to box object content property**

Note:

- **there is no inheritance**
- **using of global variables, Class methods and properties isn't allowed.**
- **box object has the following properties: height, width, length, numOfBooks, volume, material, content.**

- **The content property contains an array books**
- **book object has the following properties: title, numofChapters, author, numofPages, publisher, numofCopies**
- **you can define any function needed for both box and book objects**
- **use .toString() to tell its dimensions and how books are stored in it.**
- **implement .valueof() so that if there is more than one box object we can get total books in these boxes by adding the**
  i.e. box1 has 5 books while box2 has 2 books, box1 + box2 should return 7

3. **Create your own custom object that has getSetGen as a function value, this function should generate setters and getters for the properties of the caller object This object may have a description property of string value if needed**
   Let any other created object can use this function property to generate getters and setters for its own properties
   Avoid generating getters or setters for any property of function value

   Hint:
   if getSetGen() is applied on any other object it should generate getters and setters for all of the applied object properties
   i.e. if you have the following object
   obj = {id:"SD-10",location:"SV", addr:"123 st.", getSetGen: function(){/*should be implemented*/}}
   using of getSetGen() will generate the following getId(), setId(), getLocation(), setLocation(), getAddr(), setAddr().

If you created the following object var
user = { name: ”Ali”, age:10}

When applying getSetGen() on the user object (you can use call or bind or apply), it will result in creating the following: getName(), getAge(),setName(),setAge().