Wrapper & Iterator Lab

Due Feb 28 at 11:30am **Points** 100 **Questions** 2
Available Feb 28 at 9am - Feb 28 at 11:30am about 3 hours **Time Limit** 150 Minutes

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	104 minutes	0 out of 100 *

* Some questions not yet graded

Correct answers are hidden.

Score for this quiz: **0** out of 100 *
Submitted Feb 28 at 10:47am
This attempt took 104 minutes.

Question 1 — Not yet graded / 0 pts

Please upload a completed copy of the **Assignment Contract (https://sit.instructure.com/courses/23566/pages/assignment-slash-lab-submission-contract)** as PDF.

Per course policy - Every student must individually submit a copy of the contract, even if only one group member submits the assignment for the entire team.

If you do not include a copy of the Assignment Contract, your work will not be graded. You will be required to resubmit and the new submission will be counted as submitted at the date and time you submit this agreement.

Lab Submission Contract Lab 4.pdf (https://sit.instructure.com/files/3610723/download)

Question 2

Not yet graded / 100 pts

Write a fake vector class. The class should use the built-in vector/list/array in your language of choice to effectively "wrap" inside of it an existing vector, while presenting a limited vector-like functionality to the user. Here is a TypeScript interface that represents what your class should look like:

```
interface Vector<T> extends Iterable<T> {
 get(index: number);
 set(index: number, value: T);
 length: number;
 push(value: T);
 pop(): T;
 insert(index: number, value: T);
 // remember to implement the iterable functionality
}
```

If you are working in another language, you may translate this interface into a C# interface, Java interface, VB interface, or C++ pure virtual class, because the class you write must implement/inherit the interface/class that is shown above. If you are programming in raw JavaScript (ES6/ES2015), you cannot implement the interface, but you should carefully program your class to work identically to the presented interface.

Please ensure you implement an iterator (or Enumerable, for C# programmers). You may use generators if your language supports them, or you may implement the iterator classes manually if you choose to do so.

Note: You must not submit your "node_modules" folder if you are working on NodeJs/JavaScript. (Just submit your JavaScript source code and package.json file)

Lab4.zip (https://sit.instructure.com/files/3610721/download)

Quiz Score: **0** out of 100