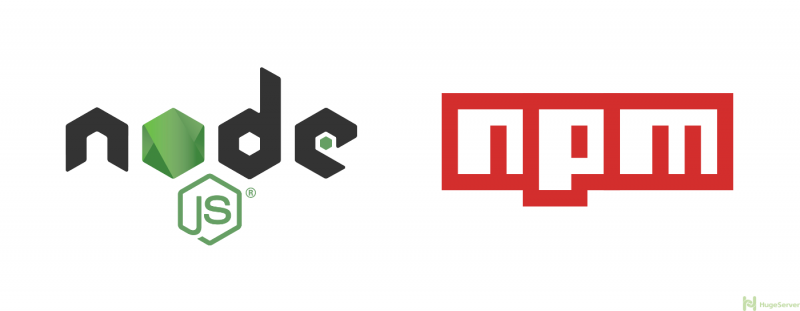
**IS 312 Web Design – TypeScript/JavaScript**

**HOP09 Hash Tables, Stacks and Queues**

6/1/2020 Developed by Amrutha Vaidyanathan

11/29/2020 Reviewed by Chuck Liu

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Understand queues in TypeScript.
* Understand stacks in TypeScript.

**Resources**

* [W3School JavaScript Tutorial](https://www.w3schools.com/js/default.asp)

**Preparation**

1. Open the VS Code, git clone <URL>(to download this Github repository to your local machine)

1. Change directory to the corresponding folder of each week. For example: Your work for module 1 should be stored under Module 1 folder; your work for module 2 should be stored under Module 2, and so on:

* cd “Module 9”

1. Now, follow the instructions provided in each folder to complete your Hands-on Practice

**Queues and Stacks**

1. Create a file called Queue.ts.



1. Compile and run the Queue files.

>>> tsc Queue.ts

>>> node Queue.js

A picture containing clock

Description automatically generated

1. Copy the Queue.ts file, rename it to Stack.ts, and update the content to match the following screenshot.



1. Compile and run the Stack files.

>>> tsc Stack.ts

>>> node Stack.js



**Linked Lists**

1. Create a LinkedList.ts file under the directory.  Add a LinkedNode class into the file.

A screen shot of a computer

Description automatically generated

Note: In this hands-on practice, we are going to create a doubly linked list.  You can try to come up with your own LinkedList<T> class utilizing the LinkedNode<T> class before continuing.  The LinkedList<T> will at least have the size(), add(), remove(), and printList() method.  Beware of the edge cases.

Method size(): Return the element count of the linked list.

Method add(element: T): Add the last element of the linked list.

Method remove(): Remove an element from the end of the linked list.

Method printList(): Print all of the elements in the linked list.

Additionally, you can also add the following methods, which we are not going to cover in this document.

Method addFirst(element: T):  Add an element to the front of the linked list.

Method removeFirst(): Remove the first element of the linked list.

Method getFirst(): Return the first element in the linked list.

Method getLast(): Return the last element in the linked list.

Method clear(): Remove all of the elements from the linked list.

Method contains(element: T): Return true if the linked list contains the input.

Method get(index: number): Return the element at the specified position.

Method set(index: number, element: T): Set the element at the specified position.

1. Add a LinkedList class with the size(), add(), remove(), and printList() method in it.

A screenshot of a cell phone

Description automatically generated

A screen shot of a computer

Description automatically generated

1. Add the code that tests the linked list.

A picture containing clock, black, sitting, white

Description automatically generated

1. Compile the TypeScript file and run the JavaScript file.

>>> tsc LinkedList.ts

>>> node LinkedList.js



**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hitting the control + ~ key and type the following command:

**>>> git add .**

**>>> git commit -m “Submission for Module 9– YOUR NAME”**

**>>> git push origin master**