PROGRAMMING ASSIGNMENT 3

CS1410 - 100 points

After you finish this assignment, you will be able to do the following:

- Define pointers
- Define structures
- Use **new** and **delete** operators
- Use the **->** operator to access structure members via pointers
- Pass function parameters by pointer
- Format output using setw

DESCRIPTION

A stack is a container whose items are organized in **Last In First Out (LIFO)** format: items that are inserted (or pushed) first into the stack are extracted (popped) last from it and items that are inserted last are extracted first. A stack typically defines two basic operations:

- The PUSH operation inserts a new item at the top of the stack.
- The POP operation extracts (removes) the item at the top from the stack.

Create a program that implements and tests a **stack** data structure that can store an unlimited of **double**-valued items. Every item in the stack must be an object of the following structure:

```
struct Item{
   double value;
   Item* next;
};
```

where the **value** member of an item contains its actual double value and the **next** member points to the item next to (or underneath) it. Your program must also use the following structure to represent the stack:

```
struct Stack {
   Item* top;
   int size;
};
```

where the **top** member of a stack always points to the item at the top of it, and the **size** member saves the number of current items in it. In the beginning, the stack will be empty which means that **top** should be **nullptr** and **size** should be **0**. Your program must also define the following three functions:

- **void push(Stack* stack, double num) -** Uses the **new** operator to create an item whose **value** is the argument **num**, inserts the created item at the top of the stack; and increments stack **size** by 1.

- **`double pop(Stack* stack)`** - Removes the item at the top of the stack and returns its **`value`** and decrements the stack **`size`** by 1. If the stack is empty, it returns the constant **`HUGE_VAL`** from the **`<cmath>`** header file. It uses the **`delete`** operator to return the popped item’s memory back to the operating system. ***Don't access the value of the item after you delete it***.
- **`void printStack(const Stack* stack)`** - Prints the values of the stack items to the console, one item per line, top to bottom. The output should indicate where the top of the stack is and how many items are there. Use **`setw`** for this. Your output should be similar to the sample outcome below.

In the **`main()`** function, define a stack variable and test it by calling the above functions to push, pop, and print out arbitrary double values.

For instance, the following sample output is obtained by pushing the values **`20.5`**, **`60.004`**, **`10.08`**, **`40.3`**, and **`-160.93`** to the stack one at a time and then calling the **`printStack()`** function:

```
-160.93 <= TOP
   40.3
  10.08
 60.004
   20.5
     -- 5 items found
```

Notice that the first pushed number is at the bottom while the last number is at the top. Notice also that the top of the stack is marked with **`<=TOP`** and the bottom is indicated by **`--`**.

Calling the **`pop()`** function once should return **`-160.93`** and change the stack so that, when **`printStack()`** is called, we get the following outcome:

```
   40.3 <= TOP
  10.08
 60.004
   20.5
     -- 4 items found
```

HINT: Check out Worksheet 6 to learn about pointers and [Example 6.2: Doubly Linked Lists]() for a similar but more complex use of pointers.

INSTRUCTIONS

For this assignment, you need to have a GitHub account. If you don’t have one already, please sign up for one at https://github.com/.

Getting the assignment starter code from GitHub:

- Sign in to GitHub.
- Go to the assignment link https://classroom.github.com/a/7ebFqKEh and accept the assignment. This should create a private repository under your GitHub username for this assignment. Click on the given link to open this repository and see the starter code.
- Click on the **Clone or Download** button dropdown and copy the given URL.
- Navigate to your assignments folder (or any folder you want this assignment to be placed in) and open it using Visual Studio code.
- In Visual Studio Code, open a new terminal and then run:

  ```
  wsl                         (for Windows 10 only)
  ```

  ```
  git clone THE_URL_YOU_COPIED
  ```

  This will download the starter code of this assignment from GitHub and create a folder for it with a name like **`cs1410-assignment-03-github_username`**. This is the folder where your program file(s) (**`.cpp`** and/or **`.h`**) should reside.
- Open the assignment folder (whose name looks like **`cs1410-assignment-03-github_username)`** in Visual Studio Code and start writing your program.

Compiling your C++ program:

- From inside the assignment folder in Visual Studio Code, open a new terminal and run:

  ```
  wsl                         (for Windows 10 only)
  ```
- To compile your program run:

  ```
  make
  ```

  This command will call the C++ compiler on your program, compile it, and, if no compilation errors are found, create an executable program named "**run**" for it. If there are compilation errors, read the console error messages and then go back to your source files (.cpp and/or .h) and fix them. Save your changes and run the "**`make`**" command to compile the program again.
- To run your program, run:

  ```
  ./run
  ```
- To clean (remove) old compilation files and start over, run the command:

  ```
  make clean
  ```

  You can now run the "**`make`**" command to compile your program again and the "**`./run`**" command to run it.

Submitting your program to GitHub:

- Make sure to save your changes and commit them to GitHub when you are done. You can do that by running the following commands from inside your assignment folder:

  ```
  wsl                         (for Windows 10 only)
  ```

```
git add .
git commit -m 'short commit message goes here'
git push
```

Make sure to do this at least once by the deadline. For your final submission, I recommend using "**Final submission**" for the commit message. **Note that committing changes is not enough; you have to push them to GitHub; otherwise, your changes will stay on your local machine and I will not be able to see your submission.**

- Go to your assignment repository in github.com and make sure your changes are there.
- Click on the **Clone or Download** button dropdown and copy the given URL. Go to Canvas and submit the copied URL. **This URL must be submitted in Canvas after you make your "Final submission" to GitHub.**

RUBRIC

CRITERIA	POINTS
Define pointers	10
Define structures	10
Use **`new`** and **`delete`** operators	10
Define and call functions with pointer parameters	40
Use -> to access structure members	10
Format output using **`setw()`**	10
Readable, commented, and properly indented code	10
TOTAL	100