Homework #4
Stack

Due: October 31 by 11:59:59 pm **Assigned:** October 17, 2018

You will create a stack. It would be wise to utilize your linked list from Homework 3.

Assignment Requirements

- The name of your final executable shall be called `hw04`
- Your source code, object files, and exectuables shall be organized
  - See the *Preparing and Submitting* section
- Download the header file from Blackboard
  - **DO NOT:**
    - Change the name of the class
    - Change the name of any function
  - **DO:**
    - Apply `const` where it should be applied
    - Add any extra private functions and data that may be beneficial

Class Requirements

- Use the Album Class available on Blackboard (will be handy for testing emplace)
- Apply `const` on all applicable return types and member functions
- See the `Stack.hpp` file for requirements of a function
- Consult the `Stack` class for further clarification on functions
- You are free to implement the class “as you see fit”
  - Obviously, requirements must still be met
  - This relates more to your “underlying” data structure, and how you choose to build the Stack (inheritance vs. composition)
  - It also relates to the helper functions you decide to use

main.cpp Requirements:

- Test your class thoroughly
- Test files will be made availabe during the last week of the assignment period

Hints:

- IMPORTANT: Because we are dealing with a template class, testing is just a little different
  - The linter will not catch as much as it usually does
  - This is due to the runtime nature of templates
  - A function won't really show that it's broken until you try using it and then compiling
- Thorough testing is extremely important
- All the exceptions required to be thrown can be found in the `<stdexcept>` library
- Consider whether functionality should be written in your `Stack` class or your underlying data structure

Reminders:

- Include a makefile!
- Be sure to include a comment block at the top of every file with the required information
  - Refer to the General Homework Requirements handout on Blackboard
- Provide meaningful comments
  - If you think a comment is redundant, it probably is
  - If you think a comment is helpful, it probably is
  - Remember that you are writing comments for other programmers, not people who know nothing (obligatory Jon Snow) about coding
- There will be no extensions

Preparing and Submitting:

- Your code must be able to compile and run on the the EECS Linux Cluster
  - You are responsible for testing your code
  - "But it runs fine on my machine!" will **not** earn you any points after the fact
- Submit **ONLY** source code files and your makefile
- You will submit a zipped tarball (*.tar.gz file) of your assignment
  - The structure of your assignment shall follow the example below:

```
hw04.tar.gz
├── src
│   └── All your source files go here
├── dbg
│   └── Your debug *.o files and debuggable hw04 will go here
├── rel
│   └── Your release *.o files and release hw04 will go here
└── makefile
```

NOTE: You do not have to include the dbg and rel folders in your tarball, BUT if you don't, your makefile should be able to generate them

- Your submission must include a makefile
- Submit **ONLY** code files and a makefile as a tarball
  - From inside your project folder, run the following command:
    `$ tar -czvf hw04.tar.gz src makefile`
- Design your makefile so that the executable file is called `hw04`
- Homework submission will be handled exclusively through the `handin` tool in the Linux Cluster. You may submit your homework using the following command:
  `~cs400/bin/handin 4 hw04.tar.gz`