- Tutorials may be submitted up to a week late (the precise deadline is given on Gradescope).
- If there are specific instructions for making a function, please follow them exactly. That means that
  - function names
  - function return types
  - parameter types and order

  should all be **EXACTLY** as described. If the script can't read it, you will receive 0 for that part.
- Your **Tutorial 5** code will be marked by a Python script running on the Gradescope server. You are being given a similar script so that you may make sure your code runs correctly.

1 Submission Instructions

Download `tutorial5.zip`. Unzip it into your working directory. There is a directory `tutorial5` and the test file `t5test.py`. In the `tutorial5` folder are `test.cc`, `defs.h`, and `Makefile` to get you started. To the `tutorial5` folder you should add the following files.

1. Header and source files for the `Date` class from Assignment 2, Section 6.1.
2. Header and source files for the `Photo` class from Assignment 2, Section 6.2.
3. Header and source files for the `PhotoArray` class from Assignment 2, Section 6.3.
4. Header and source files for the `Album` class from Assignment 2, Section 6.4.

Once you have written the classes and completed your tests, submit this tutorial to Gradescope. The Gradescope server is flexible in how you submit. You may zip the folder, zip the files, or submit the folder itself or individual files. What follows are one set of instructions that will work on Gradescope. You will zip the "tutorial5" directory into a file "tutorial5.zip". If you are doing this in the course VM you must do this from the command line. Open a terminal in the folder that contains "tutorial5". Use the command `zip -r tutorial5.zip tutorial5`. This will zip the `tutorial4` folder, or update it if you change the contents. DO NOT USE tar FILES. These do not seem to work well with Gradescope. Submit to Gradescope by the deadline. You will receive your mark immediately, and you may submit as many times as you like. You may also submit late tutorials, but there will be a penalty (somewhere between 10% and 50%). Presently the server is set to receive tutorials up to a week late.

2 Testing Your Tutorial With `t5test.py`

`t5test.py` iis a test script that is very similar to the script that is being run on Gradescope (there might be slightly different input or different even tests). So the mark you see here should be close to the mark you will receive on Gradescope. To run `t5test.py`, open a command line in the directory containing `t5test.py`. You may have to make it executable, so type `chmod +x t5test.py`. You may run the script as is, in which case it will look for a file to unzip. Or, if you have not zipped your files yet you may supply a `-nozip` argument, in which case it looks for the `tutorial5` folder.

To have the script unzip `tutorial5.zip` and then test your code, run `./t5test.py`. To skip the unzip step use `./t5test.py -nozip`. When your tutorial is being officially marked you SHOULD NOT SUBMIT A .tar FILE. Either submit a zipped file or drag and drop the tutorial5 folder into Gradescope.

Running this script will generate a file `results.txt` just outside of the `tutorial5` folder. This will have some useful output as well as the mark.

3 Learning Outcomes

This tutorial will test the dynamic memory management (destructor and deep-copy) of the `Album` class.

4 Instructions

4.1 Overview

In this tutorial you will complete two more classes from Assignment 2: the `PhotoArray` class (Section 6.2) and the `Album` class (Section 6.3). In addition, if you have not yet completed it, you will also complete the `Photo` class from Section 6.1. There is one test file provided, `test.cc` that will test the functions that you provide from those classes. You are provided with a Makefile - make any changes you see fit (and remember that Linux is case sensitive!). As usual the test script, `t5test.py` is provided. This script is run with `valgrind`, so it will check the `valgrind` output to see if there are memory leaks.

4.2 Date Class

Include the header and source files of the `Date` class from Section 6.1.

4.3 Photo Class

Complete Section 6.2 in Assignment 2 if you have not done so already. Include the `Photo` header and source files in your `tutorial5` folder.

4.4 PhotoArray Class

Complete Section 6.3 in Assignment 2. Include the `PhotoArray` header and source files in the `tutorial5` folder.

4.5 Album Class

Complete Section 6.4 in Assignment 2. Include the `Album` header and source files in the `tutorial5` folder.

4.6 Makefile

A Makefile is provided for you, but you may change it or use your own. Your Makefile should compile five object files, `Date.o`, `Photo.o`, `PhotoArray.o`, `Album.o` and `test.o`. It should link these object files to the `test` executable. In addition your Makefile should contain an `all` command that creates the `test` executable and a `clean` command that removes all executables and object files.

4.7 t5test.py

Run this python script to test the functions described above. Correct all errors.