Overview: In this lab, you will use a test-driven development approach to write implementations of two classes, one of which inherits from the other.

Specifications:

- The header files for the two classes are provided on the course web page as `Person.h` and `Student.h`. Notice that `Student` inherits from `Person` (see line 7 of `Student.h`).
- You may use whatever instance variables (and types) you deem appropriate in the header files. C++ API classes are certainly acceptable.
- You are not permitted to change the names, return types, or the parameters (types, order, number) of any methods already defined in `Person.h` or in `Student.h`.
- You must use a test-driven development approach (see `https://en.wikipedia.org/wiki/Test-driven_development`) in writing your code. More specifically:
  1. Start by writing all methods as stubs in the corresponding `.cpp` files. For any `void` method, there should be no lines of code in the body of the method. For any non-`void` method, simply return a bogus value of the appropriate type.
  2. In your tester file, write a test case for the first method you want to implement. Your test should include `cout` statements clearly indicating what you are testing, what the expected output should be, and then the result showing that (eventually) your newly-implemented method passes that test.
  3. Compile and run your tester, demonstrating that the new test fails.
  4. Write the code to implement the method.
  5. Compile and run the tester again, showing that the new test now passes.
  6. Repeat the process starting at step 2 above, until all methods have been appropriately implemented and all tests pass.
- You must work in pairs on this assignment. You should submit one piece of work for your pair. **Make sure to include both of your names in your commenting.**

Pairs: You have been assigned at random into a pair, announced in class.

Writing Your Tester: Below is an example of the type of output, and level of detail, you should be using when constructing your tester, test-by-test.

```
// ...

cout << "Testing setters/getters:" << endl;
cout << "-------------------------------------------------------" << endl << endl;

cout << "Person p;" << endl;
Person p;

cout << "p.setFirstName(\"Lilly\")" << endl;
p.setFirstName("Lilly");
cout << "First Name:    " << p.getFirstName() << "[Expected: Lilly]" << endl;

// ...
```

Submitting: Package your `*.h` and `*.cpp` files (for both classes and for your tester) into a gzipped tarball named `cmsc240_lab10_`*`netid1_netid2`*`.tgz` and drop into the `lab10/` folder in the shared Box folder of one of the persons in your pair. **Your submission is due by 23:59:59 on Sunday 8 April.**

In addition to a single tarball for your pair, you must each individually submit to your own `lab10/` Box folder a text file name `cmsc240_lab10_`*`yourNetid`*`_eval.txt` in which you indicate the following:

- Your name
- Your partner's name
- An evaluation of your partner's work as one of:
  - Did more than fair share of work
  - Did fair share of work
  - Did less than fair share of work
- A brief description of the work performed by each of you justifying your evaluation category above.