COMSC-200 Lab 3
Writing Classes

GOOD PROGRAMMING PRACTICES *show / hide*

ABOUT THIS ASSIGNMENT
In this lab session, you will further modify the **GeometryHomework** solution from labs 1 and 2 so that it uses C++ classes. You will also practice programming with classes by doing exercises from our textbook.

As you complete this assignment, post the required files to the COMSC server. You may post them all at once, or one-at-a-time as you complete them, as you prefer. The individual labs are graded in order, starting with "a", which must be fully correct before "b" is graded, and so on. This assignment is worth 50-points, to be awarded after all labs are successfully completed. Use the "Submit your SP2015 work" link on the class website to post your file for this lab, using the **lab3** folder provided.

Lab 3a [`SalesPerson.h` and `SalesPerson.cpp`]
Write the class as presented in figures 9.7 and 9.8 of our Deitel textbook, with modifications per the universal requirements listed in lab 1a, and the grading policy. Name the files **SalesPerson.h** and **SalesPerson.cpp**, and name the functions exactly as Deitel does. *Fully test* your class to assure that it is error-free and ready for distribution, but do *not* submit your test program(s).

Modify the class so that it does not read numerics directly from cin, but reads numerics into strings and converts them. Remember to use the proper includes!

The following sample code shows how to organize this multi-file project. Again, you'll submit the first two files and keep the 3rd to yourself -- the instructor has his own test CPP!

Example.

SalesPerson.h

```
// SalesPerson.h, by Joe Student (1234567)
#ifndef SalesPerson_h
#define SalesPerson_h

class SalesPerson
{
  ...
};

#endif
```

SalesPerson.cpp

```
// SalesPerson.cpp, by Joe Student (1234567)
#include "SalesPerson.h"

...
```

test.cpp: do NOT submit

```
#include <iostream>
using std::cout;
...

#include "SalesPerson.h"

int main()
{
  SalesPerson salesPerson;
  ...test all getters and setters
}
```

HINT #1: Start writing the contents of the class' H and CPP file, saving and recompling as each item is added. Also call the functions in the test CPP using the object created in main, to make sure they work right.

HINT #2: To test this one, write and call the getSalesFromUser function first. Then write and call the printAnnualSales function, and run the program to see if the output is right for the input you give it. Then write and call the setSales functions a few times to change some of the months' input values. Then call printAnnualSales again to see if the results are what you expect.

*Post **SalesPerson.h** and **SalesPerson.cpp** to the COMSC server for credit.*

Lab 3b Writing With Classes [`GeometryHomework3.cpp`]
In this rather small change to the GeometryHomework program from lab 2, replace the `struct` definitions with `class` definitions. Do *not* use more than one source file -- that means no **.h** files, and just a single **.cpp**.

Make the data members private, and provide constructors to initialize all data. Each constructor is to take a `const char*` array in its parameter list, representing an array of strings -- the tokens from the input file. This is to allow objects to be created with statements such as this: `Square* s = new Square(token);`, where `token` is the tokenized array of strings read from the input file. (The first string is the name of the shape.) So the `atof` function calls will now be made in the constructor functions and not in main's EOF loop.

Make the output functions members of the classes. As such, they no longer need names such as `outputCube` -- simply name them `output`, so they all are named the same. We will make use of this when we get to polymorphism in lab 12. The functions should no longer have a parameter -- use casted void* pointers to call these output functions with the arrow for member selection.

Note that the ONLY members of your classes should be the 1-parameter constructor, the 0-parameter void "output" getter function, and the dimensions -- nothing more than this. Do NOT store the shape's type in the shape objects -- they "know" what they are.

Post ***GeometryHomework3.cpp*** *to the COMSC server for credit.* Save this file, because modifications will be made to this program in labs 4, and 11-13. The GeometryHomework series is the subject of seven (7) lab assignments.

How to pause a console program: *show / hide*

GRADING POLICY *show/hide*