Name: ____

Problem	Points	Score
1	75	
2	25	
Total	100	

Notes:

(1) For this exam you are allowed to open a terminal window on your computer, you are allowed to web surf with Google, but you cannot use online chat or other interactive services.

(2) The first step in this exam is to create a workspace in the following directory:

/data/courses/ece_1111/current/exams/ex_03/lastname_firstname

(3) Set the permissions using "chmod -R u+rwx,g-rwx,o-rwx <lastname_firstname>" so only you have read and write permission to this directory. Create two subdirectories within this directory: p01_v01 and p01_v02. Put ALL your code in these directories. Do not touch your files after the exam is over.

As explained in class, for this exam, you need to follow our standard approach to writing C++ code. Your solution should include a make file, a header file containing your class definition, a driver program (p01.cc for problem 1) and an implementation file. Data in your classes should be "protected". Member functions should be public. You do not need any private functions for this exam.

You must complete problem no. 1 before you can proceed to problem no. 2. Problem no. 1 should be put into a directory /p01_v01. When you proceed to problem no. 2, create a directory /p01_v02, copy your code from /p01 and extend it as requested. To save you time, you do not need to rename p01 for problem no. 2.

Problem No. 1 (75 pts): You probably noticed that in America, we celebrate Thanksgiving by serving an elaborate meal that often includes turkey and pumpkin pie. Create a class called Dinner that has, as internal protected data, two classes: Turkey and PumpkinPie. The Turkey class should have internal data that consists of a float containing its weight and a character string pointer containing the vendor's name (e.g., Butterball). Your PumpkinPie class should contain the weight of the pie as a float, and the diameter of the pie in inches. You can use one header file for all three classes.

Your Dinner class needs to have four methods:

(1) Constructor that accepts values for the weight and vendor's name for the Turkey object that is data within the Dinner class, as well as the weight and diameter of the PumpkinPie object. It should pass these values to the objects contained within the class (it should initialize them via their constructors).

(2) A Destructor that cleans up memory and prints a message that it is cleaning up memory.

(3) A print method that displays all internal data (it will simply call the Turkey and PumpkinPie print methods).

(4) A debug method that prints "hello class" to stdout when called.

Your driver program should take the following arguments:

```
p01.exe 21.0 "Butterball" 2.0 9.0
```

where the first two are associated with the Turkey class and the second two are associated with the PumpkinPie class.

The driver program should create a Dinner object using the command line values, and then call your print method to display the internal data values. This should demonstrate that you have successfully constructed an object and passed the values to the objects within the Dinner class.

Problem No. 2 (25 pts): Add a static variable, static long dbg, to all three of your classes, and a method set_debug() that sets this value from the command line (by adding it as an additional argument to the command line):

```
p01.exe 21.0 "Butterball" 2.0 9.0 27
```

Note that "27" is the value you should set dbg to in this case. This variable must be declared as protected – you cannot make it public.

Demonstrate by adding augmenting your print methods, and modifying your driver program, that this variable's value is shared by all three classes. Demonstrate that when you set it anywhere in your driver program, all three copies of this variable instantly change. You should only have to call one of your class methods to set dbg. If done correctly, the other dbg variables will be set when anyone of them is changed.