CS 151 – Lab 11

Inheritance Lab

# Assignment 1 - Multiple Inheritance & Franken Classes

In our lecture, we discussed the concept of inheritance and showed how to make a derived class from a base class. In this lab, we will be exploring a concept known as multiple-inheritance where a class has more than one base class. While java does not support the concept of multiple-inheritance, your old pal C++ sure does.

A comparison between multiple-inheritance and a chain of inheritance is shown in Figures 1 and 2 below.

**Figure 1 -Multiple Inheritance (Class C inherits from both classes A and B)**

**Figure 2 – A Chain of Inheritance**

# Part 1 – Creating a Date Class

Create a class called Date. It should have the following integer attributes:

* day
* month
* year

Make sure to make these member variables protected so that they can be inherited, but not modified outside of the class.

The class should also have a constructor that accepts three integers for day, month, and year. You may assume that these will be acceptable values.

Finally, your class should have accessor functions for day, month, and year that return integer values:

* getDay( )
* getMonth ( )
* getYear ( )

# Part 2 – Creating a Time Class

Now you need to construct a class called time. Time should have the following integer attributes:

* hour
* min
* sec

As with the date class, make sure that your member variables are protected.

You should have a constructor that accepts three integers for hour, minute (min), and seconds (sec). You may assume that these will be acceptable values.

Finally, your class should have accessor functions for hour, minute, and seconds that return integer values.

* getHour( )
* getMin( )
* getSec( )

# Part 3 – Creating a Franken Class

Now that we have created a date class and a time class, we can combine them into a class called DateTime.h. The header file for this new class is posted under examples on Laulima although feel free to create your own version.

# Part 4 – Implementing the Franken Class

In this part of the lab, you will implement the Franken class, date time. You will need to make an implementation file called DateTime.cpp. This file will need to include string and should make use of the to\_string() function.

# Part 5 – Using your new class

Create a short program that uses your original date and time classes as well as your new Franken class.

# Part 6 – Object Oriented Questions

1. Why don’t all object oriented languages support multiple-inheritance? What can go wrong with multiple inheritance?

It’s not considered “safe”, it’s possible to wind up calling a function that exists in more than one place and get an error. There is also the slicing problem, where data is lost.

1. Inheritance is just one way to relate two classes. Another technique for relating two classes is known as composition. Briefly describe how composition works.

By including the class as a variable and passing the variables by references. It looks a lot like it uses pointers.

# Part 7 – Virtual Problems

Sometimes you are designing a program that needs to be versatile and handle various situations. You may not be able to account for all possible situations. For example, you have a program that is supposed to track sales, but you cannot anticipate all types of sales; regular sales, sales with discounts, mail orders, sales with shipping charges, etc. In a simple sale the bill is very simple while in more complicated ones, you may have several different things appearing on the bill. You may also want to do some other things with the sales that were made, such as finding the largest or the smallest sale.

Download the sales program from Laulima (sales.zip). Modify the definition of the class Sale by deleting the reserved word **virtual**. Before you run the program try to determine the output. Then compile the pieces and run the program.

1. Did you have the right answer after removing virtual?

No

1. What happened? Explain how the compiler handled the situation.

The compiler viewed it as the functions being exactly the same and not two different items.

1. In object oriented design, you may hear the term abstract class. What is an abstract class?

A base class with a virtual function

**Submit your code and answers to the discussion questions using the Laulima link.**