

CSE 2012 Fundamentals of Computing II

Spring 2008

Lab handout for Week of February 25

Objectives

1. Complete problems on templates
2. Prepare for the midterm exam
3. Develop an inheritance hierarchy and corresponding implementation
4. Utilize polymorphism
5. Have fun!

Pre-lab assignment

1. (5 points) Problem 14.3 in D & D.
2. (5 points) Problem 23.12 in D & D (22.12 in 6th edition)
3. (5 points) Answers to the ten sample questions prepared by the instructor and TAs.

In-lab activities

1. (1 point) Report to lab **on time**. Attendance will be taken at the scheduled lab time.
2. For this lab, you will be working in teams of 2-3 students. You may choose your final project partner(s) or another classmate in your lab section for this assignment.
3. Last Monday, you developed a UML hierarchy for a base student class and at least two levels that inherit from this base class. There is a software tool called DIA (<http://live.gnome.org/Dia>) that we will use later in the semester, so you should briefly familiarize yourself with it today. It should be installed on the boxes in the lab room; just open it and dabble for 5-10 minutes.
4. Implement most if not all the basic functions for this base class. You are required to at least implement a generic “print” function that displays student information.
5. Now implement a few functions of each of the derived classes and appropriate data members. Inside of these functions include another print function that is specific to the type of student represented by the derived class.
6. Provide a small driver program that will display the contents of a derived student class using the base class pointer. Then, display the contents using the derived class print function. Make the output slightly different so the TAs can tell there is a difference between these functions during grading.

7. Copy your current files into a new directory. Make the copied student class abstract by turning its print function into a pure virtual function.
8. Providing print functions in the derived class will make the class concrete. If you have not done so already, implement simple print functions for all derived classes that inherit from the base student class.
9. Now, modify the original driver program to utilize polymorphism to display the contents of at least three derived classes based on your abstract class, using a base class pointer. Again, modify the output such that the TAs can tell the difference between different types of students.
10. (4 points) Flag down the lab TA and have them examine and check off your work.

Post-lab (due by the last lab after spring break)

Write and submit (in your dropbox) a lab report with the following sections.

- (5 points) Code for your toy abstract student class and at least three derived classes that contain the implementations for the pure virtual function.
- (5 points) A brief report (approximately one page) describing the class hierarchy you developed, and how inheritance and polymorphism “improved “ your code. Valid answers (with descriptions of course) are it made your program simpler, fewer lines of code, easier to make additions, etc. In short, why inheritance and polymorphism are powerful C++ tools that promote reusability and modularity. Please attach the UML diagram for your student classes to this report, submitting a paper handout to the TAs if needed.

Code is expected to be well-commented, and the narrative portions of your report must be written professionally (*i.e.*, complete sentences, correct grammar and punctuation, consistent tense and voice, formal tone).