COMP 210	Object-Oriented Programming

LAB 3 & HOMEWORK 3

Abstract

In the third lab and homework you'll be *refactoring*[1] your design from homework 2 to include the use of abstract classes and class extension. You can use your solution to homework 2 to complete this assignment.

Lab

Your goal for the lab is to redraw the UML diagram[2] for a new class hierarchy design that lifts all of the shared media file implementation into an abstract class called *AbstractMediaFile*. In lab 2 I did not have you restate all interface methods in each class, but as our class hierarchies get more complicated it will be clearer to do so. Here's the rule of thumb: restate all Media interface methods in the class in which it can/will be implemented. Anything implemented in *AbstractMediaFile* doesn't need to be restated in the concrete Media types. Put another way, you should propagate interface methods down the hierarchy until it's implemented.

If by the end of lab you haven't shown the instructor your updated diagram, then do so to get lab credit[3].

Homework

Now refactor the the code to meet the new design. Do not modify your lab 2 code directly, but instead work off a copy of the project. To copy an existing project in Eclipse you can highlight the project, hit Ctrl-C to copy, then hit Ctrl-V to "paste" the project. You'll be given the option to rename the project at this point[4]. In theory, you should not need to modify any of the tests for the concrete Media types as the change that we're making is to the implementation and not the overall interface.[5]. If and when tests need to be rewritten, then you should think about whether or not you're re-factoring isn't also redesigning the overall interface. The example code from lecture notes 5 should act as a good guide for this exercise.

Submit as *hwk3* via handin no later than the next lab.

---

[1] https://en.wikipedia.org/wiki/Code_refactoring
[2] a PDF of the UML diagram from lab 2 is available on the server: /home/comp210/s18/lab2.pdf
[3] You should probably be checking your progress as you go though...
[4] You might still want to rename packages
[5] we're changing *How* not *What*