SE (M) 2017-2018 Lab 5: Design Principles

Revision: v20180502a

We recommend collaborating in pairs.

Overview

You've been hired by the startup *ExamMarkr* to refactor their existing code base. They shipped their product for enrollment season, but in the rush to get the product shipped the code is terrible to work on. Now they're struggling to add features that let you add a student's mark for a particular course. You'll first identify the poor choices made by the existing developers, refactor the code so that it is easier to maintain, then add in this missing functionality.

Identify coupling and cohesion

Read through and analyze the code. Try to identify where there are examples of coupling and cohesion. In particular, note which type of coupling and cohesion is present in the existing code. You may find it useful at this stage to quickly sketch out a UML diagram, but don't spend time making this look pretty.

Plan the solution

Now that you've noticed some problems with the existing code, plan out what needs to be changed in order to reduce coupling and increase cohesion. Again, drawing out a quick UML diagram is useful. Are your planned changes invoking a particular design pattern?

Refactor the code

Now refactor the code to your design. This may take quite some time, so don't get too bothered if you don't finish it in the lab.