Refactoring

# Strategy Pattern for ArrayLists

In the TaskList and ListOfCourses controllers there was code to convert the ArrayList of Courses or Tasks returned by the Database module to a String array to be displayed on the page. This code was transferred into a static class called ArrayConverter using one or two methods: convertCourses or convertTasks. These methods then convert the ArrayList they are passed into an ArrayList of ListItems and return a call to convertListItems. ConvertListItems converts the ArrayList into the needed String array by calling getName() for each of the items in the ArrayList.

The reason for this was to reduce the amount of work done in the controllers without changing the signature of around a dozen methods in the SQLDatabase and StubDatabase classes. The purpose of the controllers is to operate the page they refer to, and by creating the ArrayConverter the controllers do not violate the Single Responsibility Principle.

# Strategy Pattern for Grade Calculation

We implemented the Strategy Pattern again for the Grade Calculation. Initially it was the responsibility of the Task Detail controller to calculate the total grade and remaining weight available for the course being displayed. However, in order to follow the Single Responsibility Principle it was necessary to again move this code a static class called Grader.

# Abstracted Database Layer

We implemented an interface called Database. It was written in the vaguest way possible back in iteration one and the controllers used to access the database directly using staticDB. Now the controllers only use the interface instead.