CS250 UML Diagram Relations

The UML Diagram describes classes and how they relate to one another for the course management system, similar to Canvas. It includes the following classes:

Course: represents a course/class and has members such as **courseId**, **name**, **description**, **schedule**, **prerequisites**, and **instructor**. It has functions which can enroll/drop students, and get the list of enrollments.

Roster: represents the enrollment of a student in a course, with members such as **enrollmentId**, **courseId**, **studentId**, and **status**. It has a method which gets the enrollment status and returns a string with the status.

Teacher: represents an instructor with private members such as **instructorId**, **name**, and **email**

Student: represents a student with private members **studentId**, **name**, and **email**

Assignment: represents an assignment for a course, with private members such as **assignmentId**, **courseId**, **title**, **description**, **dueDate**, and **maxScore**

Grade: responsible for storing and managing the grade information for a student in a course. It contains private member variables for the **studentID**, **course ID**, **homework**, **test scores**, and **assignment grades**. The class has public methods which can set and get the grades for each type of assessment, or calculate the final grade. Furthermore, the class has methods to set the grade for a specific homework, test, or assignment, and to retrieve the current grade for a given assessment type. Lastly, the class also has a method to calculate the final grade for the course based on the total weighted average of assignment types.

Curve: responsible for implementing grade curves in a course. Has private member variables for the **curveType** and **curveAmount** along with a public function to apply the curve when given a list of grades.

Class Relations

Roster → Course: Students who enroll in a course will be enrolled on the roster for that specified course. A course can have many rosters, but each roster is only for a single course.

Course → Student: Similar relationship to the former; a student can have many (unique) courses, however the same student cannot be enrolled in the same course twice.

Course → Teacher: Each course is only assigned one teacher, but a professor can teach many different course types/sections

Assignment → Student → Course: Each Student can have multiple assignments, but each is linked to the course they are in. I.E. there will never be an assignment for a class you are not enrolled in

Assignment → Teacher → Course: Each Teacher can have multiple assignments given out, but it will always be for a specific course. I.E. Teachers will never release an assignment for a different course.

Assignment/Curve → Grade: Assignments are directly responsible for all the grades in a course and cannot be affected in any other way. Similarly, the curve will impact the grade class (this is similar to a class curve, rather than an assignment curve)