John Malandrakis CUS1156

Lab 2

Part 1

1. Class comment: This represents the information and grades for a single student

finalAverage() comment: Computes the weighted average for the student

1. The instance variables are firstName, lastName, ssn, year, finalGrade, midtermGrade, homework1Grade, homework2Grade, and homework3Grade. These variables represent the grades and personal information for the student.
2. The ClassRoster class represents a group of StudentRecords, or in plain terms, the students in a class.
3. The first lab used a CatManager class which was similar to ClassRoster because it would manage the Cat objects in a similar way that ClassRoster manages the StudentRecord objects.

Part 2

1. StudentRecord would need to be modular by using an ArrayList to add and remove assignments and exams freely. Hardcoded variables would be replaced with an ArrayList for exams, and an ArrayList for homework assignments.
2. Unused methods in my completed StudentRecord class include getFirstName, getLastName, and getYear.
3. ClassRoster and ClassRosterTester would have to do null checks and other error checks to verify that a method or output can be used. Without error checking, the compiled classes could have runtime errors such as NullPointerException or semantic errors such as negative grades (This is based on how I built the methods).
4. Code Snippet:

/\*\*

\* Returns a class average for the specified exam

\*

\* @param exam The exam ("midterm" or "final")

\* @return The class average

\*/

private double calcExamClassAverage(String exam) {

double average = 0.0;

for (StudentRecord record : students) {

average += record.getExamGrade(exam);

}

return average / students.size();

}

1. Use a single method called addHomework(String name, double grade) which can add new assignments in a modular way.
2. The primary adjustment would be to completely get rid of the hard coded homework methods as mention in the previous question. Next on the list is proper error checking for invalid or undesired values such as negative grades or incorrect ssn/name/year. The final adjustment is fully documenting the code.