For this lab assignment, you will be writing a few classes that can be used by an educator to grade multiple choice exams. It will give you experience using some Standard Java Classes (Strings, Lists, Maps), which you will need for future projects. The following are the required classes:

1. **Student** – a Student object has three private instance variables: lastName, a String; firstName, a String; and average, a double. It has accessor methods for lastName and firstName, and an accessor and mutator method for average. It has two constructors: a default constructor that creates a default Student, and one that has two String parameters to initialize firstName and lastName. The initial average of all Student objects is 0.
2. **UnitTest** – a UnitTest object represents an exam for a class of Students. It has two instance variables: input, a Scanner object, and answers, an ArrayList of Strings that represents the correct answers on the exam. In the constructor, you must read from the file “answers.txt” which represents the correct answers for this exam. You must write one method in this class: calculateGrade, which has an ArrayList of Strings as a parameter. This ArrayList has all of the Students answers for their exam. To grade the exam, you compare the answers and, if they are equal, count it as correct. The method returns a percentage, as a double in the range [0.0...100.0].
3. **UnitTestRunner** – A UnitTestRunner is the runner for the lab. There is a lot of leeway as to how you can define this class. It needs to do the following:
   * Create a UnitTest object.
   * Read in the Student information from the file allExams.txt. The file is of the format last name, first name, and answers for their exam (always ten letters).
   * Create Student objects by using the last name and first name String objects read from the file.
   * Add the Student objects to an ArrayList of Students.
   * Add all answers to an ArrayList of Strings.
   * Add these answers to a HashMap. The HashMap is of type <String, ArrayList<String>>. The String key must be the concatenation of a Student’s last name followed by their first name, with no spaces and in lower case. Hence, the String for John Smith would be smithjohn. The value to put into the HashMap is their ArrayList<String> answers. For example, the method call myMap.get(“weirrichard”) would return an ArrayList<String> object containing all of my answers for the exam.
   * Grade all of the exams. Go through the ArrayList of Student objects to get the answers from the HashMap, and use the calculateGrade method from the UnitTest object you created to grade the Student’s exam. **Use the mutator method for average in the Student object to save the grade as the average. \*\*\*YOU NEED THIS FOR LATER \*\*\***
   * Print all of the Students’ names, in the form first name and last name, separated by a space, and their grade after.

A very important aspect of this course is the creation of software that is easily extendable and robust. Although we all know that the exam from the file has ten questions/answers, your submission should work for any multiple-choice exam, regardless of the number of answers/questions. I would also like you to implement one of the following:

1. Calculate the range and mean grade for the entire class. You can find the range by finding the minimum and maximum grade (the range is minimum – maximum). The mean is the average grade for the entire class. Print this information after the students’ information.
2. Calculate the percentage of students that got each question incorrect. This would help teachers realize which topic they need to focus on or review the most for a midterm/final. This work should be done earlier in the program, ideally when you are grading exams. You can store this data in a data type of your choice: arrays, Lists, etc. Print this information after the students’ information.

Your submissions must have the three classes, as well as output. Do not worry about a UML diagram for this assignment. We will be covering composition and aggregate relationships in UML diagrams soon.

Note: Make sure all files used in this assignment are in the same directory as the source files!