Lab 5: GradeBook Object Oriented Programming

0. This lab is due by **4pm on Wednesday, June 15th.**

1. Please read instructions carefully. In this lab you will implement objects representing Students and a Course. You will be provided with a test code program called GBProgram that will interface with your objects.

2. Write a class named **Student** that implements the following Application Programmer Interface (API). You must abide by this API in order for GBProgram to properly test your code:

Class Student

Students are constructed by giving the student's name to the Student(String) constructor. Student have a running point total, initialized to zero, that is increased using the addPoints(int) method. This value is returned by the getPoints() method.

The Student class overrides Object.toString() and returns output of the form: name points

Constructor Summary
Student(java.lang.String name)
Creates a new student with the given name.

Method Summary
void addPoints(double newPoints)
Adds the given number of points to this Student's total.
double getPoints()
Returns the total points this student has earned.
String toString()
Returns a String representation of this Student of the form *name points*.

3. Write a classed named **Course** that implements the following API. Again, you must abide by this API for GBProgram to properly test your code:

Class Course

Courses are constructed by passing them an array of Students enrolled in the course. At any time, the course can return the average student score using the average() method. New scores are added to an enrolled student's total using the checkOff(Student, double) method. Finally, the report() method displays a report to the console of the form student.toString() Pass/Fail, where the student passes if they have a score higher than the average.

Constructor Summary

Course(Student[] students)

Creates a new Course that the given students are taking.

Method Summary

double average()

Returns the current average total score over all the students enrolled in this course

boolean checkOff(Student student, double score)

This method takes a Student and a score as input. If the student is enrolled in the course (that is, they were in the array passed to the constructor), then this method increments that student's point total and returns true. Otherwise, if the student is not enrolled, we return false.

void report()

Prints a report of the following form to the screen:

```
-----------
Student.toString() Pass/fail
Student.toString() Pass/fail
...
Student.toString() Pass/fail
```

In other words, this method outputs a separator, then each Student's name followed by whether they passed or failed the course on a separate line. Students will pass the course if their total is higher than the Course average.

4. **Extra Credit:** Modify your **Student** class so that students may be enrolled in multiple courses. Change **Course** so that Courses now have names like "Economics" or "Literature". Add a method *void reportCard()* to **Student** that displays all of that student's course names and whether they are passing the class, where "passing" means above the class average.

5. **Check Off:** Compile and test your program on your own. Call a staff member to your computer and demonstrate a successful run of the code. We may modify data values or quiz you on Object Oriented Programming concepts from Lecture 7 before checking off you lab.

Contents of GBProgram.java:

```
class GBProgram {
    public static void main(String args[]) throws Exception {
      Student[] students = {
          new Student("Alice"),
          new Student("Bob" ),
          new Student("Carol"),
          new Student("David"),
          new Student("Eunice")};

      Course aiti = new Course(students);

      double[][] studentScores = {
          {25.5, 39, 99},
          {29, 60, 10, 100},
          {100.0},
          {40, 22.3, 44.1, 10.2, 33.2, 19.5},
          {33.0, 67.9, 22}};

      for (int i=0; i< students.length; i++)
          for (int j=0; j<studentScores[i].length; j++)
           aiti.checkOff(students[i], studentScores[i][j]);

      aiti.report();

      System.out.println("Got: " + aiti.average() +
                         ", Expected: 150.94");

      Student alice = students[0];
      aiti.checkOff(alice, 50);
      System.out.println("Got: " + aiti.average() +
                         ", Expected: 160.94");

      Student frank = new Student("Frank");
      if (!aiti.checkOff(frank, 100))
          System.out.println(frank + " not enrolled.");
    }
}
```

Output of GBProgram:

```
Course Average: 150.94
Alice 163.5: Pass
Bob 199.0: Pass
Carol 100.0: Fail
David 169.3: Pass
Eunice 122.9: Fail
-----------
Got: 150.94, Expected: 150.94
Got: 160.94, Expected: 160.94
Frank 0.0 not enrolled
```

EC.S01 Internet Technology in Local and Global Communities
Spring 2005-Summer 2005