

Lab 6: GradeBook Object Oriented Programming and ArrayLists

0. This lab is due by **12 noon on Wednesday, June 22nd**. You should be checked off by 12 noon. Do not wait till 11.30am on Wednesday before trying to get checked off.
1. This lab is a continuation of Lab 5. In this lab you will incorporate ArrayLists for Student and Course objects in your Gradebook application. Please ensure that you have been checked off for Lab 5 so that any errors within your lab will not be carried over to this lab.
2. The course class should be modified as follows:

Class Course

Add a new field '*students*' to store a list of pointers to each student in the course. Add a new field '*studentGrades*' to hold a list of point totals for each student in the course. Add a field for the course name, '*courseName*'. If you finished the extra credit section of Lab 5, this is almost complete. Modify the constructor to include the name of the course.

Constructor Summary

Course(String courseName)

- Creates a new Course that the given students are taking.
- Initialize the ArrayList of students.
- Initialize the ArrayList of student grades.

Method Summary

String getCourseName()

- Returns the name of the course

void addStudent(Student stud)

- Adds a student to the course
- Adds the course to the student as well
- What else do you need to add? Think about retrieving grades of students

void addPoints(double points, String name)

- Adds points to the correct running point total in the '*studentGrade*' ArrayList. You must first find the student in the '*students*' ArrayList so you can add the points at the appropriate index in '*studentGrade*'.

double studentScore(Student studentName)

- Returns the total points for the student in the course.

double averageAll()

- Returns the average of the total points of all the students in the course.

void report()

Prints the name of each Student in this Course, and whether they are passing or failing.

Students will pass the Course if their total points are higher than the average of all students taking this Course.

3. Modify the Student class of lab 5 is to include a record of the courses each student is taking. You do not need to create new files for this class. You can just add on to the class from Lab 5.

Class Student

Add a data member, called '*courses*' that holds the Courses that a student is enrolled in. This should be an ArrayList. Use the appropriate access modifier.

Constructor Summary

Student(java.lang.String name)

Creates a new student with the given name.
Initialize the ArrayList.

Modify the methods:

Method Summary

void addPoints(double newPoints, Course courseName)

Adds the given number of points to the points for a specific course. Remember that you can call the addPoints() method in the Course class.

double getPoints(Course courseName)

Returns the total points this student has earned for the course specified

void addCourse(Course courseName)

Adds a course to the Student's list of courses.
Be sure this doesn't create an infinite loop when called from Course.

void getCourses()

Prints the names of the courses the student is taking

String getName()

Returns the name of the student

String toString()

Returns a String representation of this Student of the form:

StudentName

CourseName Pass/Fail

CourseName Pass/Fail

...

All the courses the student is enrolled in should be printed along with the student status in the class (ie. Passing/failing)).

4. Write test code in the main program GBProgram

- Create 3 new students – Chilongo, Gikandi and Amwayi
- Create 4 courses – Accounting, AITI, Information Systems, and Economics.
- Enroll each students in at least 3 courses
- Assign points to each student. Call the addPoints method more than once.
- Invoke the report method on all the courses
- Invoke the toString method on all the students
- Change the courses and grades
- Invoke the report method again

5. **Checkoff**

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 your test code and quiz you on various aspects of your lab.

We will not accept modifications to the required files after the lab is officially due.