

Introduction to Computer Science Lab 11 – More Array Concepts

Name _____ Section _____

This lab covers some of the important array concepts in Java. You should print a copy of this lab so you can record your answers to questions. Turn in the completed questions and any supporting printouts as required in the lab. Here is a link to the API documentation (<http://java.sun.com/j2se/1.5.0/docs/api/index.html>) in case you need it.

P1: Parallel Arrays - Create a folder and project and copy this class into a new file (<http://cs.uakron.edu/~margush/209Labs/code.htm>). This program stores course grade information in parallel arrays. Read through the program, make sure the program runs, and then go on to the next step.

Oh - you need a data file! You should create a file in the same folder as the program's class files. Name it ... well, you can read the code to see what it needs to be named! You can paste the contents from here (<http://cs.uakron.edu/~margush/209Labs/data.htm>).

Complete the method named **gpa()** to calculate (and return) the GPA. The GPA is the weighted average of the course grades. For each course, calculate its quality points (grade*credits). Sum these to get total quality points. Divide by the total number of credits. You should use the **numericEquivalent** method to convert the letter grade to a number. Read that function and then explain how it works:

Explain:

Why is this method (**numericEquivalent**) declared **static**?

Answer:

The use of parallel arrays is difficult because it relies on every section of code that alters the list to be extremely careful to keep the arrays synchronized. If the data for a class is to be removed from the collection, you must be careful to remove the corresponding grade and credit data, and slide data in the same way in all the arrays. If any of the data gets out of order, the association between the data in the arrays is lost.

Implement the **remove** method that searches for a course name (provide this as an argument to the method) and removes it from the structure if found. Uncomment the statements in **main** that will remove the first and last classes in the list and then redisplay the course data and gpa. Check that the remove method works correctly.

Tell the GPA displayed before and after the remove:

Print your completed file (be sure your name is in the file).

This program only handles data for one student. Consider the difficulty in expanding this parallel array structure to handle more than one student. Would we create an array of parallel arrays? Hmmmm... sounds like fun. Read on!

P2: Replacing Parallel Arrays - Create a new file containing a class named **Course**. This will store the name of a course, the credit hours and the letter grade for the course. Provide public accessors and a constructor that requires all three items to be specified when a **Course** object is created. Include a public method that can return the quality points for a course (credit hours*numeric grade). This method should use the **static** method copied into the class described in the next paragraph.

Make a copy of the original class (renaming it appropriately), and then modify it to use a single array of **Course** objects to hold course data and calculate GPA's. The **main** method should not be altered in any way (except for the name of the class). Test that everything works as before, including the **remove** method. You will need to make modifications in nearly every method.

Reflect on the simplifications you have made. How does the elimination of parallel arrays through the introduction of a new class simplify the list management algorithms?

Observations:

Print the two files for Part II. Be sure your files contain your name and the current date. Submit your solutions (3 files) in a single stapled package together with this completed document.