**CS201 Lab 11**

**40 points**  **Due**: 04/21/15

**Problem:** You previously wrote a program to calculate simple statistics about recent earthquakes (PA3). However, you now need more complicated problems solved that require you to store the information in the program and then process it later.

**Purpose:** This lab gives you practice with:

* Programming classes
* Using classes with data members

**Details:**

In this lab we will solve the rest of our earthquake processing problem. At this point you should have an Earthquake class with the appropriate data members and constructor, and an EarthquakeProcessor class that can read the data from the file and store it in an array of Earthquakes. Your Lab10.java file should call that method to successfully read from the file. Your UML should include method headers for the problems you are solving this week.

This week you will build on the code you wrote last week. If you are unsure about the code you used in Lab10, or if you were unable to complete it successfully, you may copy the sample solution in the Lab10Solution repository to your Lab11 repository, instead of using the code you wrote for Lab10.

In this lab you must complete the following:

* Adding appropriate accessors and mutators to the Earthquake class
* Adding a compareTo method to the Earthquake class
* Writing a method in EarthquakeProcessor for each problem that must be solved: output all of the earthquakes to a new file in sorted order (ask user for the name of the file), output all earthquakes with a magnitude equal to or higher than a magnitude given by the user, and output the largest depth of all earthquakes.
* Completing the main java file to have a working menu, and that asks for the necessary user input for each problem and then calls the appropriate methods in EarthquakeProcessor.

In this lab you will need to use an EarthquakeSorter class to sort your earthquakes. This code has been provided for you in your Lab11 repository.

**Steps:**

1. As a class we are going to discuss the idea of a compareTo method, which you need to be able to sort the earthquakes.
2. Decide what code to build on for Lab11 and copy it to your repository (either your Lab10 code, or the Lab10solution code). Also copy your UML diagram, as you will have to complete it this week as well.
3. Complete your Earthquake class and make sure it compiles:
   1. Add accessors
   2. Add mutators
   3. Add compareTo method as discussed in step 0
4. Add to the EarthquakeProcessor class:
   1. Add one method at a time for the 3 tasks you must solve. Remember that the earthquakes you are interacting with are in the array that is a data member of the class.
5. Complete the parts of the Lab10 class you have not yet completed:
   1. Complete the menu method
   2. Complete the outputAnswers method. NOTE: There is an error in the original method header. You need to have an EarthquakeProcessor parameter for the method to work!
6. TEST that it all works!!!
7. Add/modify methods to your UML diagram as necessary.
8. Write comments in your code, and at top of each new method to make it clear what it is doing. You should already have comments at the top of each class from Lab10. If you use the sample solution you will need to add appropriate comments; I purposefully did not do that for you.

**Submit:**

1. To GitHub:
   * All Java files & UML diagram file
2. On paper in class:
   * A hardcopy of the classes you wrote code (1 copy per pair)
   * A short (250 words or less) individual reflection about what you learned, and how pair programming worked for you this week. Also discuss what it was like to build one larger program across 2 weeks, instead of the normal 1 week labs.