**CS201 Lab 10**

**40 points**  **Due**: 04/14/15

**THIS LAB CANNOT BE TURNED IN LATE**

**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. *We will solve this problem across two labs.*

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

* Programming classes
* Using more than 1 class in your program

**Details:**

In this lab we will solve the first part of our problem: Designing our classes, and designing/writing our program to read in the earthquakes file, store it in an earthquakes object, store all earthquakes in an array. You have an initial UML diagram in a separate Word document in your repository. You will need to complete part of this UML diagram today, and code part of two classes. The rest we will finish in Lab 11. You should reference your or your partner’s PA3 code and use it to help you complete this lab.

In EarthquakeProcessor we will store the array of Earthquakes as a data member. Since we don’t know how many items are in the file, we create an array of size 10000, and when we read in from the file we need to count how many are actually in the array to use later when we calculate statistics.

**Steps:**

1. Create your Earthquake class:
   1. Think about what data needs to be stored about each Earthquake, and add the necessary data members and constructor to the UML diagram for the Earthquake class. Data members are listed as “– name:type” if they are private.
   2. Get Dr. Olsen’s approval for your Earthquake design.
   3. Create a new Java file to represent a single Earthquake, and write the code to match your UML diagram. Compile it.
2. Add to the EarthquakeProcessor class:
   1. The constructor and data members are already written for you.
   2. Write the method to read from the file. See the comments in the method for your algorithm.
   3. Make sure it compiles.
3. Complete the parts of the Lab10 class you can complete at this time to run your program:
   1. Write the announce method.
   2. Complete the main method:
      1. You need to ask the user for the name of the file to read from, create an EarthquakeProcessor object, then call the readFromFile method.
      2. You need to call the menu and do the chosen option until the user choose to quit by calling the outputAnswers method. So far they have no choices!
   3. Compile and run, make sure it does not crash.
4. Start planning your methods for next week. The tasks you need to solve are to output all of the earthquakes to a new file in sorted order, to output all earthquakes with a magnitude equal to or higher than a magnitude given by the user, and to output the largest depth of all earthquakes. Add each method to the UML diagram.
5. Write comments in your code to make it clear what it is doing.
6. Write comments for each method in your code, and for each class.

**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.