DEPARTMENT OF PHYSICS & ASTRONOMY
PHAS3459 Mid-Term Exam
18th November 2015
10:00 – 13:00

Please read the exam guidelines, rules, instructions and marking criteria at https://moodle.ucl.ac.uk/mod/page/view.php?id=2016117 (linked from the PHAS3459 Examinations Moodle page).

This exam is worth 25% of your final mark for the course and is made up of two parts:

- 15 multiple-choice questions, worth 7.5% of your final mark;
- a programming exercise, worth 17.5% of your final mark.

The multiple-choice questions are given as a Moodle quiz.

The duration of the exam is 3 hours. For the programming exercise, you will write Java classes and methods to read data from a URL, analyse the data and present the results. The Java source code of your solution to the programming exercise should be uploaded using Moodle under the section headed “Exam I”.

Each class should be uploaded as a separate file. Your classes should be created in a package called “`exam1`”. This package must contain at least one “`main`” class called “`MidTermExam`”: you should also include any other classes you create. You must upload all your classes used in your solution, including any you have copied or imported from earlier coursework modules. The code you upload **must be self-contained**: the marker must be able to compile and run it using only the classes uploaded and the Java API. If you use your own classes from earlier modules, make sure you copy them into the `exam1` package and upload them along with any new classes you create during the exam.

You are advised to read the entire exam paper before starting work.

DEPARTMENT OF PHYSICS & ASTRONOMY
PHAS3459 Mid-Term Exam
Programming Exercise

Background

Earthquakes are recorded by many organisations around the world as a way of cataloging existing seismic activity and helping to predict future serious earthquakes. In this exercise, you will analyse some selected data from a database of earthquake events in one region of the Earth with well-studied seismic activity from the year 1989.

The data you will use is at the URL
http://www.hep.ucl.ac.uk/undergrad/3459/exam-data/2015-16/earthquakesCA1989.txt.

The first two lines are header lines, containing the name of each field in the data. Each of the remaining lines contains the following details of one earthquake event, separated by whitespace:

- 3 columns giving the year, month and day (GMT) of the earthquake (YYYY MM DD);
- 3 columns giving the hour, minute and second (GMT) of the earthquake (hh mm ss.sss);
- 2 columns giving the latitude and longitude of the earthquake in degrees (LAT LON);
- The depth of the earthquake in km (DEP);
- 3 columns giving the horizontal major/minor axis [km] and azimuth of the 95% relative location error ellipses that define how precisely known the location of the earthquake is (EH1 EH2 AZ);
- The error on the depth of the earthquake measurement in km (EZ);
- The magnitude of the earthquake (MAG);
- The unique ID of the earthquake measurement (ID).

The data from the 4 error values is missing for some measurements: this is indicated by a value of -1 in each column.

Tasks

You should write a program using appropriate classes and methods to read the data from the URL given above, store the data in suitable collection objects, and carry out the following tasks. When printing details of the earthquake, you should not include data from any of the 3 location error measurements (EH1, EH2 or AZ).

- Print the total number of earthquakes recorded in the file.
- Print details of the earthquake with the largest magnitude.
- For each month, print:
    - the number of earthquakes measured;
    - full details of the deepest earthquake;
    - full details of the earthquake whose depth is most accurately known *ie*. has the smallest depth error (for those where this data is given).

Marks will be awarded for the following:

- Correctly loading and parsing the data from the given URL;
- Organising the data using appropriate `Collection`(s);
- Correctly identifying the total number of earthquakes and the earthquake with the largest magnitude;
- Sorting the data correctly and storing the sorted data using the appropriate `Collection`;
- Correctly printing the correct details for each month, as given above;
- Appropriate use of `Exception` handling;
- Well commented code that is clearly structured and laid out and follows the Java indentation conventions.

More information on the marking criteria can be found at https://moodle.ucl.ac.uk/mod/page/view.php?id=2016117.

END OF PAPER