**CSE210 Advanced Object Oriented Programming**

**2018 Coursework**

***Deadline****: 16:00, 2nd, May, 2018*

**1. Overview**

The objective of this coursework is to develop a practical application for data processing and simple analysis based on the object-oriented principles you learn from this module.

**2. Description**

***2. 1. Dataset***

The dataset consists of records of researchers crawled from the website ResearchGate (<https://www.researchgate.net>, thanks to Mr. Hang Dong, PhD candidate at XJTLU, for sharing the dataset). In total, there are more than 6,000 researchers and topics+skills. Each record (row) contains information about a researcher; for this coursework, you only need to focus on the following columns:

* university
* department
* user (i.e., his/her full name)
* topic and skill (together they are referred to as research interests or interests). Interests under these two columns are delimited with a comma (,).

***2. 2. Data Processing and Analysis***

The dataset is stored in an EXCEL file and needs to be processed from your Java program. You can use the Apache POI (the Java API for Microsoft Documents) library for this purpose. The API is available at:

<https://poi.apache.org/download.html#POI-3.17>

You need to download it and put it into your classpath in order to use it. A tutorial on how to use the library is here:

<http://poi.apache.org/spreadsheet/quick-guide.html#CellContents>

You need write your code to extract information as specified in Section 2.1. You also need to write code to perform some simple data analysis, e.g., how many researchers have the same interest, how many times two research interests co-occur, or what the similar interests are for a given interest.

***2.3. Objects***

It is important to follow good object-oriented programming principles and practice, e.g., object modelling, functionality decomposition, code reuse. At the same time, you need to ensure correctness and take robustness and efficiency into account.

You need to design a number of information objects to represent the objects together with the methods for this application. You need to design a number of helper classes in order to realise the functionalities (as specified in Section 3). A separate class for testing purposes is also needed. Below is an example about the objects for this coursework; however, you are welcome to have your own design following good programming practice and style.

* Researcher – represents the relevant information about a researcher;
* Interest – represents a research interest, e.g., either a topic or skill in the dataset; note that a researcher might have multiple interests.
* LinkedList – represents a list of objects, e.g., research interests of a researcher should be stored in a linked list.
* Utility and helper classes – as needed for data processing and analysis.
* Test class – for testing the implemented functionalities.

**3. Tasks**

You should complete the following tasks.

* **T1**: develop information objects and helper classes (with their methods) needed for this application with good coding style, and design an easy-to-use interface for testing (command window is enough).
* **T2**: consider coding style, efficiency and robustness (e.g., exception handling and data structures).
* **T3**: complete the following tasks which will be checked and marked during the demonstration sessions. You should design a method for each of the tasks (but that method may call other methods).

**Note: FAILING TO PRESENT AT THE DEMONSTRATION SESSION WILL RESULT IN ZERO MARK FOR THE DEMONSTRATION PART (SEE MARKING SCHEME).**

* *T3-1*: calculate the number of distinct researchers in the dataset.
* *T3-2*: calculate the number of distinct interests in the dataset.
* *T3-3*: given a researcher’s name, show detailed information about him/her (e.g., university, department, interests).
* *T3-4*: given an interest, calculate the number of researchers who have that interest.
* *T3-5*: given two interests, show the number of times they co-occur.
* *T3-6*: given a researcher, find similar researchers based on their interests. This could be used in expert recommendation applications. It can be done using different methods, here are some examples: (1) cosine similarity (<https://en.wikipedia.org/wiki/Cosine_similarity>), (2) clustering, you need to self-study some clustering algorithms and choose one to use. The book [1], especially Chapter 6.6, is a good one for you to get started; any other fundamental textbooks on data mining and machine learning are also fine. The Weka library is recommended for your implementation, <https://www.cs.waikato.ac.nz/ml/weka/documentation.html>, <http://weka.sourceforge.net/doc.dev/>), (3) probabilistic topic models see reference [2] for details, the Mallet API from University of Massachusetts, is recommended, <http://mallet.cs.umass.edu/api>. Marking for this task will be based on the quality of the research, techniques and algorithm you use, as well as the produced results.
* **T4**: use Javadoc comments to document your codes and generate HTML javadoc. For task T3-6, you should have a detailed documentation on the algorithms or techniques you use.
* **T5**: demonstrate your work by yourself in the same order as specified in the lab group file (note that time for each demonstration varies and you probably need to wait a bit longer than expected). You will be asked to modify your code during the demonstration (to prevent plagiarism). The ability to modify code is measured in %. Marks for demonstration will be scaled by the %. Being able to modify code according to requirements gets 100%, not able to modify anything (will report for potential plagiarism) will get 0%. Other cases will be considered accordingly. Feedback will be provided during the demonstration.

**4. Deliverables**

You should deliver **ONE FILE** (via ICE) according to the following description.

* All source codes, compiled classes and any other supplementary files regarded necessary.
* HTML documentation. This can be generated by using either the javadoc utility or the plug-in in your IDE.
* A separate **README** file explaining how to run your code, e.g., path for the dataset, how to install external libraries, and how to test all the functionalities.
* **DO NOT** include the external APIs in your submission as the size exceeds the file limit on ICE.
* *Submit all the source codes, compiled classes, documentation and any supplementary files regarded necessary, in* ***ONE ZIPPED FILE***.

**5. References**

[1] Ian Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, <http://cs.du.edu/~mitchell/mario_books/Data_Mining:_Practical_Machine_Learning_Tools_and_Techniques_-_2e_-_Witten_&_Frank.pdf>

[2] Steyvers, M., & Griffiths, T. (2007). Probabilistic Topic Models. In T. Landauer, D. McNamara, S. Dennis, & W. Kintsch (Eds.), Latent Semantic Analysis: A Road to Meaning. Hillsdale, NJ: Laurence Erlbaum. <http://psiexp.ss.uci.edu/research/papers/SteyversGriffithsLSABookFormatted.pdf>