**Assignment 1**

**Project Description**

Assignment Goal: Develop a program, using Java, to assign courses to students based on their preferences.

* There are 9 courses being offered. Each course has the following information.
  + Capacity - The total number of students that be registered for this course.
  + Class Timings - For simplicity, timings are just represented as positive integers, starting from 0.
  + Course Name (A,B,C,D,E,F,G,H or I)
* The following information is associated with each student.
  + Student ID (a 3 digit integer in the range [100,999])
  + Levels:
    - FIRST\_YEAR
    - SECOND\_YEAR
    - THIRD\_YEAR
  + *Hint: Enums can be used to specify the levels of a student.*

The following rules MUST be followed when registering students to courses.

1. A student cannot be registered to multiple courses that have the same class timing.
2. A student can take a maximum of 3 courses. It is possible that some students will be assigned <3 courses as it depends on the input file.
3. All students are required to provide 9 preferences.
4. Total number of students can exceed the total capacity across all 9 courses.
5. A course is assigned to a first year student only if there is no 2nd or 3rd year student still waiting for it. Among 2nd and 3rd year students, priority has to be given to the 3rd year student.
6. If a course has been filled up, then any further registration requests for that course are rejected.
7. Each student has a satisfaction rating for each requested course. The order in which a student requests courses defines the satisfaction rating of that student for those courses given by *(9 - course.index)*.  
   For example, let **A,B,C,D,E,F,G,H,I** be the requested courses of a student. Then,  
   satisfaction rating of student for course **A** = 9 - A.index = 9 - 0 = 9  
   satisfaction rating of student for course **B** = 9 - B.index = 9 - 1 = 8  
   satisfaction rating of student for course **C** = 9 - C.index = 9 - 2 = 7  
   satisfaction rating of student for course **D** = 9 - D.index = 9 - 3 = 6  
   And so on...

***Note: The average satisfaction rating for a student determines the quality of the course registration algorithm.***

**INPUT FORMAT**

Your program should take two input files - *student\_coursePrefs.txt* and *courseInfo.txt*. Note that the **input files will be well formatted**.

*student\_coursePrefs.txt* will have the following format,

<student\_id> <PREF\_1>,<PREF\_2>,<PREF\_3>,<PREF\_4>,<PREF\_5>,<PREF\_6>,<PREF\_7>,<PREF\_8>,<PREF\_9>::<student\_level>

*courseInfo.txt* will have the following format,

<course\_name> CAPACITY:<capacity>;CLASS\_TIMING:<class\_time>

**INPUT EXAMPLES**

*student\_coursePrefs.txt*

111 D,C,A,B,G,I,H,F,E::FIRST\_YEAR  
222 F,E,D,C,B,A,H,I,G::SECOND\_YEAR  
333 D,A,F,E,I,C,H,B,G::THIRD\_YEAR

*courseInfo.txt*

A CAPACITY:30;CLASS\_TIMING:7  
B CAPACITY:20;CLASS\_TIMING:8  
C CAPACITY:40;CLASS\_TIMING:7  
D CAPACITY:60;CLASS\_TIMING:9  
E CAPACITY:40;CLASS\_TIMING:2  
F CAPACITY:50;CLASS\_TIMING:8  
G CAPACITY:45;CLASS\_TIMING:4  
H CAPACITY:25;CLASS\_TIMING:3  
I CAPACITY:10;CLASS\_TIMING:6

**OUTPUT**

Your program should write the registration results to an output file called *registration\_results.txt*.

*registration\_results.txt* will have the following format,

<student1\_id>:<course\_1>,<course\_2>,<course\_3>::SatisfactionRating=<value>  
<student2\_id>:<course\_1>,<course\_2>,<course\_3>::SatisfactionRating=<value>  
<student3\_id>:<course\_1>,<course\_2>,<course\_3>::SatisfactionRating=<value>  
...  
AverageSatisfactionRating=<value>

**Clarifications based on student questions**

* + **Q**: *Can we assume that student\_coursePrefs.txt and courseInfo.txt will be passed in as command line arguments in a specific order?*
  + **Ans**: Yes. Please make sure to include the assumed order of the input files in the README.
  + **Q**: *Can we assume that there will never be a class with multiple sections?*
  + **Ans**: Yes. Each course will have only one entry in courseInfo.txt file.
  + **Q**: *Can we assume capacity > 0? (for each class)*
  + **Ans**: Yes.
  + **Q**: *For output formatting, if a student does not get 3 classes should there be an extra comma?*
  + **Ans**: This is left to you. As nothing related to this was mentioned in the assignment, you are free to either remove the third comma or not. Please mention in the README, the formatting that you have chosen for students that were assigned < 3 courses.

**Compiling and Running Java code**

* Your submission must include a readme in markdown format with the name **README.md**.
* Your README.md file should have the following information:
  + instructions on how to compile the code
  + instructions on how to run the code
  + justification for the choice of data structures (in terms of time and/or space complexity).
  + citations for external material utilized.
* You should have the following directory structure (replace firstName\_lastName with your name).
* ./firstName\_lastName\_assign1
* ./firstName\_lastName\_assign1/coursesRegistration
* ./firstName\_lastName\_assign1/coursesRegistration/src
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/util
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/util/FileDisplayInterface.java
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/util/FileProcessor.java
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/util/Results.java
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/util/StdoutDisplayInterface.java
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/driver
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/driver/Driver.java
* ./firstName\_lastName\_assign1/coursesRegistration/src/coursesRegistration/scheduler
* ./firstName\_lastName\_assign1/coursesRegistration/src/build.xml
* ./firstName\_lastName\_assign1/README.md
* [Other Java files you may need]

**Code Organization**

* Your directory structure should be EXACTLY as given in the code template
  + Download the ANT based tarball [here](http://www.cs.binghamton.edu/~mgovinda/courses/downloads/firstName_lastName_assign1.tar.gz).
  + Use the command on linux/unix: *tar -zxvf firstName\_lastName\_assign1.tar.gz*.