

Michelle M. Khalife

Part 1 – Implementing a Simplistic Representation of the CS Department

Complete and/or edit the representation below as you see fit and draw the class diagram.

1. A course is characterized by the following attributes:

- id,
- title,
- discipline,
- numberOfHours, and
- numOfGroups

Note: Make sure there's a copy constructor for your course.

2. A professor is characterized by the following attributes:

- a 4-digit id,
- a name (last name, first name),
- a seniority level (0.00 – 60.00),
- setOfDisciplines<String>, and
- listOfAffectedCourses<Course>

Implement a professor class that implements the *Comparable* Interface and overrides *compareTo()*: the more senior professor is the one with higher seniority. If two seniorities are equal, default to the date of hire; the more senior professor is the one who was hired first.

Note: when the professor object is constructed, the `ArrayList<Course> listOfAffectedCourses` is null.

3. A department is characterized by the following attributes:

- courseMap<courseId, Course>, and
- listOfProfs<Professor>

Note: Create a partially parametrized constructor that accepts an `ArrayList<Professor> listOfProfs` and creates a new `HashMap<String, Course> courseMap`.

Part 2 - Implementing a Shallow Match Algorithm using Multiple Data Structures

There are three types of files available to you:

- a. professors.txt (1) allows you to instantiate professors and populate listOfProfessors
- b. ListeDeCours.csv (1) allows you to instantiate courses and populate courseMap
- c. {id}_request.txt (83) allows you to examine teacher requests.

Start by reading the professors.txt file and enqueueing the professors in a priority queue. Then, instantiate a cs department. And finally populate the courseMap.

Each dequeue() allows you to process a professor's request. Get the professor's id and open the corresponding file {id}_request.txt

We process one course request at a time (one line): for a teacher to obtain a course, the course must exist in courseMap, the teacher must have the discipline, the right language, and the number of hours cannot exceed the maximum per week. Pay attention to group numbers. Depending on the number of groups assigned to the teacher, decrement the numOfGroups.

Add the course (using a copy constructor) with the group number assigned to the teacher to their listOfAffectedCourses.

Document the decisions and exceptions that you are handling.

Display the course assignments when the algorithm is done processing.