**Designing a Hiring Filtering Algorithm**

Adapted from exercise by Evan Peck (Bucknell) for CSCI 203

Imagine you are working for *Moogle,* for a well-known tech company that receives tens of thousands of job applications from graduating seniors every year.

Since the company receives too many job applications for HR to individually assess in a reasonable amount of time, you are asked to create a program that algorithmically looks at applications and selects the ones most worth looking at (and passing onto HR).

It’s difficult to create these first-pass cuts, so *Moogle* designs their application forms to get some numerical data about their applicants’ education. Job applicants must enter their gender, citizenship, and the grades they received in 6 core CS courses as well as their overall course average. This is stored in an array which you can access. For example, a student who received the following scores in classes:

* **applicant ID (5 digits, unique system generated) (0 index):** 10032
* **gender (0=F, 1=M) (1 index):** 0
* **US citizen (0=Y, 1=N) (2 index):** 0
* **Intro to CS (3 index):** 100
* **Data Structures (4 index):** 95
* **Software Engineering (5 index):** 80
* **Algorithms (6 index):** 89
* **Computer Organization (7 index):** 91
* **Operating Systems (8 index):** 75
* **Overall College Average (9 index):** 83

Would result in the following array: {10032,0,0,100,95,80,89,91,75,83}

Because you are processing many applicants, you receive an *array of arrays*. For example, this would the information for 3 applicants:

­{ {10048,0,0,90,98,91,92,99,98,98}, {10049,1,0,100,100,62,94,92,97,79}, {10050,0,0,69,97,64,66,84,91,79} }

Your job is to

1. Determine how you are going to select the top applicants to pass onto HR
2. Given a list of applicant data (an array of arrays), write a function that returns a new array of worthwhile candidates.

Download Hiring.java and input.txt to get started.

Deliverables:

Today: In groups of 4 document (and email to bauerm@iit.edu) your ideas.

Next Class: Code/Test.

Third Class: Present to class your explanation of your algorithm and the reasoning behind it and any assumptions you made. Present to class your list of worthwhile candidates.