Final Project Guidelines

- All the following functions must be defined and called (may create additional functions)
    - Main
    - Process students file
        - Must open file named students.txt (all lowercase) in *read* mode
    - Process courses file
        - Must open file named courses.txt (all lowercase) in *read* mode
    - Print menu
        - 1. Add course
          2. Drop course
          3. Print student's schedule (for one specific student – ask user for student's ID)
          4. Print course schedule
          5. Plot a bar graph with unique numbers on the x-axis and capacity/actual enrollment on the y-axis
          6. Done
    - Get menu option from the user
        - User should enter a number 1-6, *not* the name of the menu option
        - Verify that the input is valid, and re-prompt if invalid using a loop
    - Print course schedule
        - Must display unique number, course title, professor, and seats available for *each* course
    - Print student information for a particular student
        - Must display student name, ID, and a list of unique numbers the student is enrolled in
    - Get student ID from user
        - Verify that the ID exists, and re-prompt if invalid using a loop
    - Get unique number from user
        - Verify that the unique number exists, and re-prompt if invalid using a loop
    - Write updated information to 2 separate files named students-updated.txt and courses-updated.txt
        - Call this function when the user is done (when they enter a 6 for the menu option)
        - Must open files in write mode, *not* append mode
        - Files must have the same format and structure as the original files
            - For students-updated.txt, ID:First name:Last name:unique numbers separated by colons
            - For courses-updated.txt, unique number;course abbreviation and number;Professor;number of seats taken;total number of seats
- All opened files must get closed
- Create a loop in main to allow the user to go through multiple menu options
- Each time a student adds or drops a course, you must update the student's list of courses *and* update the number of seats taken in the course

- Students should not be allowed to add a course they are already enrolled in (tell user the student is already enrolled, and go back to menu)
- Students should not be allowed to drop a course they are not enrolled in (tell user the student wasn't enrolled, and go back to menu)
- Students should not be allowed to add a course if there're no available seats (tell user the course is full, and go back to menu)
- Program must use dictionaries to keep track of students and courses
- Menu must re-print each time the user is prompted to enter a menu option
- Graph must have the following features
    - x and y axis labels
    - x and y ticks
    - Title
    - Legend
- Global *variables* are **not** used
- Global *constants* are defined outside of all functions for file names and menu options
- No hard-coding course or student information (must be read from the files)
- Program should work with any number of students and any number of courses
- All logic, other than call to main, must be included in user-defined functions
- Each section of logic must be commented
- **TEST YOUR PROGRAM MULTIPLE TIMES BEFORE SUBMITTING** (change the information in the text files and re-test your program – many points will be deducted if any errors occur at any point in the program run)
- NOTE: using Object Oriented Programming (and creating a Course class and a Student class) could be an efficient method for this project. However, that method isn't required. If you're turning in multiple files, please do so in a zip folder. If you completed the project in a single file, you can turn in just that one file.

Thoroughly read through the instructions document.