

# Assignment 3 – CSRS (CSU Student Record System)

Maximum Points = 50

The purpose of this lab is to focus on the study of classes, objects, inheritance and polymorphism.

The registrar is designing a new system to manage and process student records. You have been asked to write the program to implement this.

## BASIC ASSIGNMENT

- a) Design and implement a base class that represents up to 500 university students. A student has a student ID, last name, and first name. Undergraduate, graduate, and transient students also have major, hours completed, and GPA.
- b) Design and implement classes for undergraduate, graduate, non-degree, and transient students where undergraduates have a class level (freshman, sophomore, junior, senior) and transient students have a current university.
- c) Make sure to include necessary constructors, accessors & mutators (gets/sets), and toString methods for all classes.
- d) Design and implement a main class that
  - a. reads the data from a file (students.txt) where the first field identifies the type of student (U – undergraduate, G – graduate, N – non-degree, T - transient ). Each record consists of the data for the student with each field separated by #.
  - b. After reading in **all** of the data, your main class should provide a menu that allows the user to do **at least two** of the following:
    - i. Print a list of all students.
    - ii. Print a list of majors.
    - iii. Allow the user to add a student to the list of students.
    - iv. Search for a student using a student ID entered by the user.
    - v. Remove a student from the list selected by student ID.
    - vi. Allow the user to compute the average GPA for a specified major.
    - vii. Print the list of undergraduates for a specified class level (i.e all freshman).
    - viii. Compute and print the total hours for all students with a specified major.
    - ix. Print the list of undergraduates separated by class level (CHALLENGING).
    - x. Other approved option.

## Sample Input

U#000123456#Monet#Claude#Art#120#3.45#senior

G#000111222#Slick#Grace#Music#12#3.25

T#000111333#Von Neumann#John#Computer Science#45#2.12#University of Sofia

N#000999888#Newton#Isaac

(Due before class on Wednesday, February 23, 2011) Submit a .doc file containing the UML class diagram showing inheritance for all the classes used in your program. [10 pts]

(Due before class on Wednesday, March 2, 2011) Submit your .java files containing your program to the dropbox in WebCT. [50 pts]

Grades are determined using the following scale:

- Runs correctly.....:\_\_\_/10

- Correct output.....:\_\_\_/10
- Design of output.....:\_\_\_/8
- Design of logic.....:\_\_\_/10
- Standards.....:\_\_\_/7
- Documentation.....:\_\_\_/5

[Grading Rubric](#) ([Word document](#))