CSCI 23000 Computing I

**Final Project**

**Description**

You will implement a mini version of Canvas, with data stored in csv files, e.g. use one file to store usernames and passwords, one file to store courses taken by a user etc. You will implement an admin module and a student module. When the admin module runs, the user will login and be able to add students and the courses they take. The information will be saved into csv files. When the student module runs, the user will login and be able to see the courses they take (based on data entered by the admin and stored in csv files). Whenever an admin enters more information in future runs, it will append to the existing csv files.

**Requirements and Rubric**

1. Admin Module
2. Admin login control: create a csv file to store admin usernames and passwords(at least three admins). This file will be used to authenticate admins. This is the only csv file created by you. All other csv files will be generated through coding. (5’)
3. An admin will only be allowed to enter the wrong username/password combination 5 times. (5’)
4. An authenticated admin will be provided options to
   1. add a new student (information can include but not be limited to first name, last name, unique username and password). (5’)
   2. add a new course (information can include not be limited to a unique course number and course title) (5’)
   3. add a course enrollment (student username and course number, duplicate rows are not allowed) (5’)
   4. See all the student information. The output must be nicely formatted. (5’)
   5. See all the course information. The output must be nicely formatted. (5’)
   6. See all the enrollment information. The output must be nicely formatted. (5’)
5. Must ensure that usernames and course numbers are unique. (5’)
6. Information will be saved to three csv files to store student information, course information and enrollment information. (5’)
7. Student Module
8. Student login control: use the csv file generated in the admin module to authenticate a student. A student will only be allowed to enter the wrong username/password combination 5 times. (5’)
9. An authenticated student will see all the courses he/she is enrolled in. The data will be retrieved from the curse enrollment file. (5’)
10. The enrollment output must be nicely formatted. (5’)
11. For Both Modules
12. Have file(s) to store utility functions, such as user verification, login control, add a student, add a course, add an enrollment, view students, view courses, view enrollments, check uniqueness etc. (5’)
13. Object Oriented programming methods must be used.
    1. The class created must include but is not limited to Student, Admin and Course. Define meaningful methods in each class. (5’)
    2. Create a parent class called User, and inherited by Student and Admin classes. Inheritance must be well implemented. (5’)
14. File names must be meaningful and indicate the purpose. (5’)
15. Follow professional coding standards. (5’)
16. No instances of bad coding. (5’)

**Submission**

Put all your files in one folder, zip up the folder, and submit the zip file. Your zip file name must be *YourFullName*FinalProject.zip. (5’)