Lab 5: Student course management tool

Objective: Create a tool where students can add and drop classes (similar to Lionpath)

1. This Login part is based on your previous lab.

|  |
| --- |
| Lab2: User login system  Create a user login system. Your code should do the following:  1.Load your user database from “UD.txt”  2.Display “Login or create a new user? Select L to login, select C to create new user.”  3. If wrong selection is entered, take the user back to step 2.  4. If the user entered “L”, display “Please enter your user name and hit enter”  5. Check to see if the user is in UD.txt, if not, display “User not found” take the user back to 2  6. If the username is in UD.txt, then prompt user to enter password by displaying: “Enter password”  7. check to see if the user has the correct password, if yes, display “you are logged in”, else, display “wrong password” and take the user back to 2.  8. If the user entered C at step 2, prompt the user to create a new username by displaying “Please enter your first name, last name, and student ID”, separated by a space. The user name should be the First letter of the first name, the first two letters of the last name, and the first three numbers of the student ID. For example: if the user entered “Tom Waits 567890”, the user ID should be “twa567”.  9. Once username is generated, then prompted the user for password by displaying “Please enter password”  10. Prompt the user to reenter password “Please reenter password:”  11: Check to see if the two passwords are the same, if not, display: “Password did not match”  12. If two passwords match, save the new generated username and password into “UD.txt”  13 Take the user back to 2, and make sure the new username and password works. |

2. Create two class, a student class, and a course class, take advantage of the codes provided in the class slides and sample codes.

|  |  |
| --- | --- |
| Student | Course |
| \_\_Username:str  \_\_Courses:list | \_\_courseName:str  \_\_students: list |
| addCourse():None  dropCourse():None  getCourse():list | addStudent:None  dropStudent:None  getStudents():list  getNumber of students:int |

3. Create a student object for every student in the UD.txt (you can use a loop for this)

4. Create 6 course objects:

|  |
| --- |
| 1. CS131  2. CS132  3. EE210  4. EE310  5. Math320  6. Math220 |

5. After the student user login to their account as we have done in lab 2. Display the following:

|  |
| --- |
| A: Show all courses available  B: Add a course  C: Drop a course  D: Show all my courses  E: Exit |

When students choose option A, display the following, where # is the actual number of students enrolled in the class.

|  |
| --- |
| 1. CS131 students number:#  2. CS132 students number:#  3. EE210 students number:#  4. EE310 students number:#  5. Math 320 students number:#  6. Math 220 students number:# |

Let the students add or drop a classes as they wish for option B and C. When they choose E, exit the program.

6. For option D, show the courses in the student’s course list.

7. The hard part. Since your program will end when user chose option E. You need to keep track the classes been added and dropped by each students, and who are actually in the classes. Therefore, based on your experiences in lab 3 and 4, create a SI.txt (student info) to store the courses in each student’s course list. Create CI.txt (course info) to store all the students enrolled in each course. This will be executed in the background every time when the user chose option E: Exit. When you run your program, and create your course objects, this information needs to be read into each student and course object. When a student log into his or her account, the student should be able to see what courses is in the course list by chose option D. The number of students in each course also need to be displayed in Option A. It is critical that you store the course list for each user in SI.txt, and store the number of students in CI.txt. Different users will log in and check for course info, and this info has to be updated and saved after each login.

Deliverables:

1. Save your code in the following format: CS132\_Labnumber\_Last name.py