San Jose State University

Data Science Department

CheckMyGrade Application

DATA 200 - Lab 1

Professor: Paramdeep Saini

Background

The project will provide students with experience creating applications in Python. Students will use object-oriented techniques to develop a **CheckMyGrade** console-based application.

You need to develop **CheckMyGrade** app which is an evaluation app for students' grade. This project provides the functionality to develop project that can evaluate, and display students' grades based on their scores in various subjects. It will use various data structures such as Array/Link List/Stack/Queues/Hash Table to store and retrieve grades with the optimized way.

In this part of the project, you will implement the ability to develop

You need to develop **CheckMyGrade** Object-Oriented Application, and your application must use linked list or array to store the student details such as name, course and grade. The application must have capability to add/delete/modify the student records. **CheckMyGrade** app must provide functionality to display sorted data based on student name and grades/marks and it must support searching the records. You also need to print timing of searching the records. **CheckMyGrade** App must support statistics-based functions such as getting averaged score/median score of the course. This app must have functionality to show reports course wise/professor wise/student wise to display grade report. Your application must have functionality to store all this data in file which can be used for adding/deleting/reading/writing the student records. Your app must support course addition/deletion/modification, and this can also be stored in file. So, you will have 4 CSV files students/professors/course/login.

Steps

Always test as you go!

1. Identify Classes/Data Members/Member Function

Identify the classes you need in **CheckMyGrade** Applications. We are providing some of the class details, you are free to add more classes based on your application implementation.

Sno	Class	Data Member	Member Functions
1	Student	FirstName LastName email_address Courses.id Grades marks	display_records() add_new_student() delete_new_student() check_my_grades() update_student_record() check_my_marks()
2	Course	Course_id Credits Course_name	display_courses() add_new_course() delete_new_course()
3	Professor	Name email_address Rank Course.id	professors_details() add_new_professor() delete_professore() modify_professor_details() show_course_details_by_professor()
4	Grades	Grade_id Grade Marks range	display_grade_report() add_grade() delete_grade()

			modify_grade()
5	LoginUser	Email_id password	Login() Logout() Change_password() Encrypt_password() decrypt_password()

2. CSV File

CSV files are simple plain text files. Each line in the file has one record. **Each field in the record is separated by commas**. This is a common file format which can be exported or imported by most applications. It is a common format for exchanging data between otherwise incompatible systems. To process the .csv files, we will use the **csv** library. The contents of course/student/professor csv files are nothing but corresponding class data members.

Student.csv

Email_address	First_name	Last_name	Course.id	grades	Marks
sam@mycsu.edu	Sam	Carpenter	DATA200	A	96

Course.csv

Course_id	Course_name	Description
DATA200	Data Science	Provides insight about DS and Python

Professor.csv

Professor_id	Professor Name	Rank	Course.id
micheal@mycsu.edu	Micheal John	Senior Professor	DATA200

Login.csv

User_id	Password	Role
micheal@mycsu.edu	AQ10134	professor

If user enters the password Welcome12#_ while registering then in file, it must go encrypted which some random string. When User try to login, you must read this encrypted password and decrypt the password to original password.

3. Object Oriented Design (OOD) for CheckMyGrade Application

You need to design classes relationship diagram, and you must clearly show IS-A/HAS-A relationship. You can use Visio/Draw.io or any other online tool to design the class diagrams.

4. Implement CheckMyGrade Application

Implement the CheckMyGrade Application based on the OOD diagram you created at step 3.

All the records upon addition/deletion/modification must appear in csv files of corresponding entities. Also, while implementing the code, we must include following:

- Student_id/course_id/professor_id must be unique and not null.
- You must a way to search the records specified by user.
- You must have a way to sort the records.
- Password in file must be encrypted.

Note: Above mentioned point, we will be discussing in coming classes.

5. Test the Code

Now that you have implemented the back-end functionality as well as the user interface, you can test the code.

- Implement the unit Test Modules (https://docs.python.org/3/library/unittest.html) which can perform following
- Testing of student records addition/deletion and modification. Perform the test to have the student files atleast 1000 records.
- Load the data from previous runs saved in the csv files and you must be able to load the data and search. Once completed, print the total time taken in search cases.
- You must have a test case to sort the student records (ascending/descending order) based on marks or student email_address. Your report must include the timing it took to sort the records.
- Unit test to add/delete/modify the course
- Unit test to add/delete/modify professors.

6. GitHub

Create your repo on GitHub and commit your application Find your repository url on GitHub. Then navigate to your project folder and use the following commands. Replace X with the week/module number you are submitting. Replace {your url} with the address of your GitHub repository.

At the **Anaconda Prompt**, the following commands will: stage the changes, commit the changes, and push the updates to GitHub.

```
git add --all
git commit -m "Module X"
git push {your url}
```

←Change X to the Module you are submitting.

←Change {your url} to the url for your repository on GitHub.

Note: VS Code users can use the Source Control tab on the left to stage, commit, and push updates to GitHub.