Question 1 - Student Information Center [30 marks]

You are asked to create a program for a Student Information Center. This program gives these menu options to the users to choose from:

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: #

This program saves its information inside three text files: student\_info.txt, course\_info.txt and teacher\_info.txt. These three files might have existed before or they might have been new. If they exist your program will append to them, if they do not exist your program will create these files.

student\_info.txt file

The student's records in this program are stored in a text file called student\_info.txt. Each student has these details in this text file:

* Student ID
* Student name
* Program code
* Courses: including course code, course score

The format of student\_info.txt file should be:

2067564342,Albert Smith,CS123,CSC300:65,ITS230:80

Notice the ':' between course code and course score.

course\_info.txt file

The course information will be saved in a text file called course\_info.txt, which has the following details:

* Course code
* Course name
* Credit hour

The format of course\_info.txt file should be:

ITS230,Object Oriented Programming,3

CSC300,Introduction to programming,3

teacher\_info.txt file

The teacher's details will be saved in a file called teacher\_info.txt and has these details:

* Staff ID
* Staff name
* Courses codes

The format of teacher\_info.txt file should be:

1089786,Franklin Dowyer,CSC230,ITS230

You have to strictly follow the file format in this description.

**The program**

The program continues to print the above menu until the user enters zero to exit. Before the program prompt the user with the menu, it adds an empty new line before the menu. Look at the sample run of the program at the end of the description.

Each menu option in your program has to be written as a function. You have to name these functions according to this naming convention:

1. Function for adding a course (function should be called: **add\_course**)
2. Function for adding a student (function should be called: **add\_student**)
3. Function for adding a teacher (function should be called: **add\_teacher**)
4. Function for listing all students (function should be called: **list\_students**)
5. Function for listing all teachers (function should be called: **list\_teachers**)
6. Function to search for a student by their name or student ID (function should be called: **search\_student**)
7. Function for listing the teachers and their courses for a student (function should be called: **list\_teachers\_courses**)
8. Function to show the GPA of a course (function should be called: **course\_gpa**)

For example, in the menu option 1 , to add a product you are required to call **add\_course().**

The program continues to print the above menu until the user enters 0 to exit.

All of these functions do not take in any arguments or return any values EXCEPT for **student\_gpa** which takes in a list as an argument, and returns a float value.

Your program is case-insensitive.

**Please make sure your functions use the same names as specified above!**

**Adding a course** **(option 1)**

The information for courses is saved in the course\_info.txt. When the user chooses option 1 to add a course these details will be asked from them and will be saved in the course\_info.txt:

Enter your choice: #1

Enter the course code: #CSC300

Enter the course name: #Introduction to programming

How many credit hours does this course have? #3

You are writing a function called **add\_course** to run this option in the menu. That means when the user enters option 1 the function **add\_course** will be called in your code. It asks the user to enter the course code, course name, and credit hour and then add this new course to the course\_info.txt file. This function does not return any value.

If the course code exists in the course\_info.txt file (course name can be repetitive), the user will get this error and the information about the course will not be saved in the file:

This course code exists in our database.

Sample Output:

Enter your choice: #1

Enter the course code: #CSC300

This course code exists in our database.

Enter the course code: #ITS230

Enter the course name: #Object Oriented Programming

How many credit hours does this course have? #3

The credit hour for each course should be an integer between 1 and 5 inclusive.

If the credit hour entered is not valid, the following error will be shown:

...

How many credit hours does this course have? #@

The credit hour for each course should be an integer between 1 and 5 inclusive.

How many credit hours does this course have? #1.5

The credit hour for each course should be an integer between 1 and 5 inclusive.

How many credit hours does this course have? #3

...

To check if the credit hour is valid, you must write a function is\_valid\_credit\_hour , which should take one argument and return a boolean value indicating if the given credit hour is valid. This function will be later used in **Question 2 Part B** of the exam for testing.

is\_valid\_credit\_hour('3') -> True

is\_valid\_credit\_hour('1.5') -> False

is\_valid\_credit\_hour('aa') -> False

**Adding a student** **(option 2)**

If the entered choice is 2 to add a student, the program will ask for all inputs related to the student which are student ID, name, the program code, and the number of courses for this student. For each course, the user will have to enter the course code and the score for that course. You are writing a function called **add\_student** to run this option in the menu.

The details of the students will be stored in the student\_info.txt and are linked to the details of the courses in course\_info.txt. There should be an error handling placed to prevent incorrect inputs from the user (refer to the **Error Handling section** of the description).

This is the sample output of the program when the user enters 2:

Enter your choice: #2

Enter the student ID: #2067564342

Enter the student name: #Albert Smith

Enter the program code: #CS123

How many courses does this student have? #2

Enter course code 1: #CSC300

Enter the student's score for this course: #65

Enter course code 2: #ITS230

Enter the student's score for this course: #80

If the user enters a course that doesn't exist in the course\_info.txt file it will get an error:

Enter your choice: #2

Enter the student ID: #2067564342

Enter the student name: #Albert Smith

Enter the program code: #CS123

How many courses does this student have? #2

Enter course code 1: #CSC790

This course does not exist in our database.

Enter course code 1: #CSC300

Enter the student's score for this course: #65

Enter course code 2: #ITS230

Enter the student's score for this course: #80

If the student ID exists in the student\_info.txt file (student name can be repetitive), the user will get this error and the information about the student will not be saved in the file:

This student ID exists in our database.

Sample Output:

Enter your choice: #2

Enter the student ID: #2067564342

This student ID exists in our database.

Enter the student ID: #2158432198

Enter the student name: #Eric Lim

Enter the program code: #CS123

How many courses does this student have? #2

Enter course code 1: #CSC300

Enter the student's score for this course: #67

Enter course code 2: #ITS230

Enter the student's score for this course: #90

The user can enter a maximum of 5 courses for each student each time, if they enter more than 5 courses they will get this error:

...

How many courses does this student have? #6

You can enter a number between 1 to 5.

How many courses does this student have? #3

...

If the number of courses entered is not an integer, the following error will be shown:

...

How many courses does this student have? 2.5

Number of courses should be an integer value.

How many courses does this student have? a

Number of courses should be an integer value.

How many courses does this student have? 2

Enter course code 1: csc300

...

Similarly for Scores:

...

Enter the student's score for this course: -2

The score for each course should be in the range of 0 to 100.

Enter the student's score for this course: a

Score should be a float value.

Enter the student's score for this course: 30.5

...

The user can enter each course code only once for each student. If they enter a duplicate course code they will get an error:

Enter your choice: #2

Enter the student ID: #2158432198

Enter the student name: #Eric Lim

Enter the program code: #CS123

How many courses does this student have? #2

Enter course code 1: #CSC230

Enter the student's score for this course: #65

Enter course code 2: #CSC230

This course has been entered before.

Enter course code 2: #CSC149

Enter the student's score for this course: #68

These sample outputs show some of the errors that need to be handled, for the full list, please refer the **Error** **Handling section** down below in the description.

**Adding a teacher** **(option 3)**

If the user enters 3 to add a teacher the program would ask for all inputs related to the teacher which are staff ID, name, and the number of courses they are teaching. For each course, the user will have to enter the course code. You are writing a function called **add\_teacher** to run this option in the menu.

The details will be stored in teacher\_info.txt.

This is the sample output of the program when the user enters 3:

Enter your choice: #3

Enter the staff ID: #1089786

Enter the teacher name: #Franklin Dowyer

How many courses does this teacher teach? #2

Enter course code 1: #CSC230

Enter course code 2: #ITS230

If the course does not exist, the user will get this error and the information about the course will not be saved:

Enter your choice: #3

Enter the staff ID: #1089786

Enter the teacher name: #Franklin Dowyer

How many courses does this teacher teach? #2

Enter course code 1: #CSC278

This course does not exist in our database.

Enter course code 1: #CSC230

If the staff ID exists in the teacher\_info.txt file (teacher name can be repetitive), the user will get this error and the information about the teacher will not be saved in the file:

This teacher ID exists in our database.

Sample Output:

Enter your choice: #3

Enter the staff ID: #1045795

Enter the teacher name: #Simon Gray

How many courses does this teacher teach? #1

Enter course code 1: #CSC230

If the user enters a duplicate course code, they get this error:

Enter your choice: #3

Enter the staff ID: #1089786

Enter the teacher name: #Franklin Dowyer

How many courses does this teacher teach? #2

Enter course code 1: #CSC230

Enter course code 2: #CSC230

This course has been entered before.

Enter course code 2: #ITS230

The user can enter a maximum of 5 courses for each teacher each time, if they enter more than 5 courses they will get this error:

...

How many courses does this teacher teach? #6

You can enter a number between 1 to 5.

How many courses does this teacher teach? #3

...

If the number of courses entered is not an integer, the following error will be shown:

...

How many courses does this teacher teach? 2.5

Number of courses should be an integer value.

How many courses does this teacher teach? a

Number of courses should be an integer value.

How many courses does this teacher teach? 1

Enter course code 1:

...

These sample outputs show some of the errors that need to be handled, for the full list, please refer the **Error** **Handling section** down below in the description.

**Calculating GPA (Recursive function)**

The process of calculating GPA will be done during the listing of all the students. You have to write a *recursive function* called **student\_gpa** to calculate the GPA for students. Your recursive function should get a list of scores with the course code for each student from student\_info.txt with their credit hours from the course\_info.txt , and based on that calculate their GPA.

You can use this table to convert the scores for each course to "Grades" and then calculate the GPA based on the grades:

A picture containing text, screenshot, number, font

Description automatically generated

The GPA is calculated based on the sum of grades multiplied by their credit hour, divided by the total credit hours for the student:

GPA = Sum of grades multiplied by credit hours of each course / Total credit hours

The function call for the recursive function will be a list of tuples, each tuple has a course code and a score for that course. The returning value from the recursive function is a float number, showing the GPA of the student. For example, if we were to use this function to calculate the GPA of a student with the two below courses, our function call would look something like this:

'CSC200' -> 3 Credit Hours

'ITS230' -> 3 Credit Hours

ls = [('CSC200',45), ('ITS230',90)]

student\_gpa(ls) #2.665

This function can be used in the next menu option of the program when the user chooses to list all students. You may also find it useful in other options too.

When writing your recursive function identify the base case and recursive case and comment them in your code. Identifying base case and recursive cases has a portion of your mark for this question.

You might think of other non-recursive solutions which are easier to solve this task, but your program has to do this part of your code (calculating GPA) recursively, otherwise, no marks will be given to this section of your code.

**List all students** **(option 4)**

If the user inputs 4 in the original menu, they are getting a list of all students' details with their calculated GPA rounded to **2** decimal places (it would help to use the above recursive function for calculating GPA). You are writing a function called **list\_students** to run this option in the menu.

Here is how the output of the program looks like when the user enters 4:

( assume these students exist in student\_info.txt )

Enter your choice: #4

Student ID: 2089786756

Student Name: Stanely Parade

Program Code: CS456

Course Credit Hour Score

ITS780 3 85.0

ITS569 3 87.0

ITS452 2 85.0

GPA is 4.00

Student ID: 2067564356

Student Name: Alex Frenandex

Program Code: CS349

Course Credit Hour Score

CSC300 3 75.0

ITS569 3 91.0

CSC560 2 65.0

CSC650 2 74.0

GPA is 3.57

**List all teachers** **(option 5)**

You are writing a function called **list\_teachers** to run this option in the menu.

When the user chooses this option, it will get the following output from the program:

(assume these teachers exist in teacher\_info.txt)

Enter your choice: #5

Staff ID: 1089786

Teacher name: Franklin Dowyer

Teaches the following courses:

CSC300: Introduction to Programming

CSC340: Object Oriented Programming

Staff ID: 1076342

Teacher name: Dowryn Moore

Teaches the following courses:

CSC230: Statistics and Machine learning

ITS657: Introduction to Math

ITS231: Basics of Calculus

....

**Search for a student** **(option 6)**

This is the output of the program when the user selects to search for a student in the main menu. Search can be done using the student ID or name of the student. You are writing a function called **search\_student** to run this option in the menu.

Once the student ID or name has been entered, the student information will appear.

Note: GPA is rounded to **2** decimal places here as well.

( assume this student exists in students\_info.txt )

Enter your choice: #6

Search students based on:

Select 1 for ID

Select 2 for Name

Enter your choice: #1

Enter student ID: #2067564356

Student ID: 2067564356

Student Name: Alex Frenandez

Program Code: CS349

Course Credit Hour Score

CSC480 3 75.0

ITS569 3 91.0

CSC560 2 65.0

CSC650 2 74.0

GPA is 3.57

If the user enters a student ID or name that does not exist in the system, it will get this error:

This student ID/name does not exist in our database.

Sample Output:

Enter your choice: #6

Search students based on:

Select 1 for ID

Select 2 for Name

Enter your choice: #2

Enter student name: #Matt Roberts

This student ID/name does not exist in our database.

Enter student name: #Eric Lim

Student ID: 2067123356

Student Name: Eric Lim

Program Code: CS123

Course Credit Hour Score

CSC230 3 45.0

ITS230 2 90.0

GPA is 2.40

If the user enters a student name that is duplicated in the students\_info.txt file (Remember the student name can be duplicated, but the student ID has to be unique) the program will print all the records of duplicated student names from the file with their unique Student IDs.

For this example, imagine we have 2 students called "Eric Lim" in our database:

Enter your choice: #6

Search students based on:

Select 1 for ID

Select 2 for Name

Enter your choice: #2

Enter student name: #Eric Lim

Student ID: 2067123356

Student Name: Eric Lim

Program Code: CS123

Course Credit Hour Score

CSC230 3 45.0

ITS230 2 90.0

GPA is 2.40

Student ID: 2054318123

Student Name: Eric Lim

Program Code: CS123

Course Credit Hour Score

CSC480 3 75.0

ITS569 3 91.0

CSC560 2 65.0

CSC650 2 74.0

GPA is 3.57

You need to handle all errors in user inputs here as well, please refer the **Error** **Handling section** down below in the description

**List the teachers and their courses for a student (option 7)**

When the user chooses this option, the program has to look through the three available files to find the relevant information for the user. The user enters a student ID and will get the name of the teachers that teach the courses they are enrolled in.

You are writing a function called **list\_teachers\_courses** to run this option in the menu.

Enter your choice: #7

Enter student ID: #2063643557

Teacher Name: Franklin Dowyer

Course Name: Introduction to programming

Teacher Name: Deborah Roger

Course Name: Statistics and Machine learning

If the student is enrolled in a course Statistics and Machine learning and this course is taught by multiple teachers, all of them need to be printed:

Enter your choice: 7

Enter student ID: 2032154400

Teacher Name: Dowryn Moore

Course Name: Statistics and Machine learning

Teacher Name: Hailey Franklin

Course Name: Statistics and Machine learning

If there is a course with no associated teacher, they will only get the name of the course and a message like this:

....

Course Name: Basics of Calculus

The teacher for this course is not specified.

If the user enters a student ID that does not exist in the system, it will get this error:

This student ID does not exist in our database.

Sample Output:

Enter your choice: #7

Enter student ID: #2164532999

This student ID does not exist in our database.

Enter student ID: 2063643557

Teacher Name: Franklin Dowyer

Course Name: Introduction to programming

Teacher Name: Deborah Roger

Course Name: Statistics and Machine learning

These sample outputs show some of the errors that need to be handled, for the full list, please refer the **Error** **Handling section** down below in the description

**Show the GPA of a course** **(option 8)**

When the user chooses this option the program should produce the GPA for a course searched by the course ID. You are writing a function called **course\_gpa** to run this option in the menu.

The output of the program is similar to this:

Enter your choice: #8

Enter the course code: #CSC300

GPA for this course is 2.67

The GPA is calculated based on the above-mentioned formula for GPA and is considering the scores of all the students that enrolled in this course so far.

If the user enters a course code that does not exist in the system, it will get this error:

The course code does not exist in our database.

Sample output:

Enter your choice: #8

Enter the course code: #MAT998

The course code does not exist in our database.

Enter the course code: CSC300

GPA for this course is 2.67

Note: GPA is rounded to **2** decimal places here as well.

You need to handle all errors in user inputs here as well, please refer the **Error** **Handling section** down below in the description

**Exiting the program**

When the user enters zero as an option the program should exit with this message:

Enter your choice: #0

Exiting the program...

**Error handling**

All inputs in your program should be checked:

* Wrong input for menu selection, anything other than the specified menu options is not accepted. The error message is: Wrong menu selection.
* The length of the student ID should be 10 characters. The error message is: The length of the student ID should be 10 characters.
* The length of the teacher ID should be 7 characters. The error message is: The length of the staff ID should be 7 characters.
* The length of the program code entered should be 5. The error message is: The length of the program code entered should be 5.
* The length of course codes entered should be 6. The error message is: The length of course codes entered should be 6.
* The length of the student, teacher, and course name should be between 1 and 40 characters inclusive. The error message is: The length of the name should be between 1 and 40 characters inclusive.
* The student, teacher, and course names should be alphabetic. The error message is: The name should be alphabetic.
* The credit hour for each course should be an integer number more than 0 and less or equal to 5. The error message is: The credit hour for each course should be an integer between 1 and 5 inclusive.
* The score for each course should be a float value. The error message is: Score should be a float value.
* The score for each course should be in the range of 0 to 100. The error message is: The score for each course should be in the range of 0 to 100.
* Your program is case-insensitive, and if the user enters anything in capital letters or lower letters it will be treated the same.

You can assume that errors/cases not mentioned in the description will not be tested.

Errors are ordered by priority e.g Length of name would be checked before determining whether it is alphabetic or not

**A Sample Program Execution**

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: 1

Enter the course code: CSC300

Enter the course name: Introduction to Programming

How many credit hours does this course have? 3

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: 2

Enter the student ID: 2067564342

Enter the student name: Albert Smith

Enter the program code: CS123

How many courses does this student have? 1

Enter course code 1: CSC790

This course does not exist in our database.

Enter course code 1: csc300

Enter the student's score for this course: 65.5

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: 3

Enter the staff ID: 1089786

Enter the teacher name: Franklin Dowyer

How many courses does this teacher teach? 1

Enter course code 1: CsC300

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: 4

Student ID: 2067564342

Student Name: Albert Smith

Program Code: CS123

Course Credit Hour Score

csc300 3 65.5

GPA is 3.00

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: 5

Staff ID: 1089786

Teacher name: Franklin Dowyer

Teaches the following courses:

CsC300: Introduction to Programming

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: a

Wrong menu selection.

Main Menu

Choose any of these options:

1. Add a course

2. Add a student

3. Add a teacher

4. List all students

5. List all teachers

6. Search for a student by their name or student ID

7. List the teachers and their courses for a student

8. Show the GPA of a course

Enter zero to exit the program.

Enter your choice: 0

Exiting the program...

**Testing your Code**

This program does not have thorough public test cases to check all the functionalities of your code. You are responsible for testing your code thoroughly. Question 2 in this exam is about designing your test cases for this program and also implementing one of your test cases for this program.

Hardcoding to pass any of the testcases is forbidden in your code and will not receive any marks.

You cannot use the **global** keyword in your code.