Python Basics

Practice problem

Assessment Portal

Introduction

In this assignment, you are required to develop a simple assessment portal. System should allow us to add the students in the database, add the marks subject-wise and fetch the data for each of the students.

Your main program should first display a menu as the following.After selecting the available options program will go ahead and ask for further inputs.

```
==============================================================
Welcome to the Student and Assessment Management System
<Add> Enter A to add details of a student.
<Insert> Enter I to insert assignment marks of a student
<Search> Enter S to search assessment marks for a student.
<Q> Enter Q to quit.
==============================================================
```

Program Organization

1. **Add option**

   If a user chooses the option **<Add>** the program then the console should ask to enter the student ID, name of the student, and course ID. After inserting all the details for a student, inserted data will display the student ID, name, and course they are enrolled. Subsequently the data entered should be added in the **students.txt** file.

   Here you have to make sure that the student ID is unique and there is no duplicate entry.

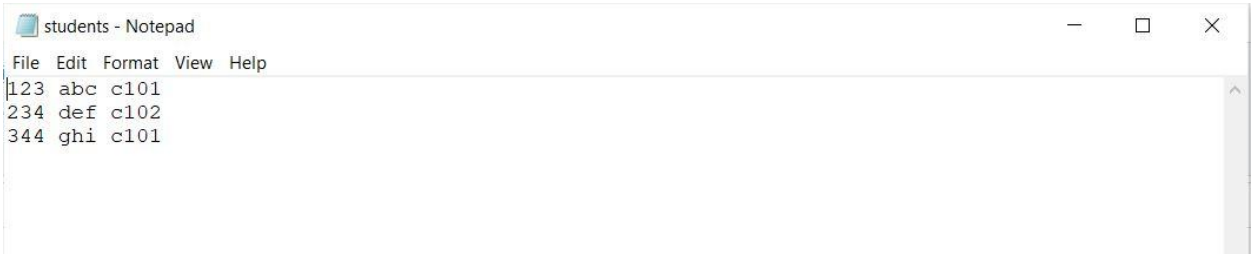
   Once the new record has been added, the system will then ask the user **'Do you want to enter details for another student (Y/N)?'**

   - If they enter **'Y'**, the system will allow them to enter details for another student as before.
   - If they enter **'N'**, the system will display the main menu again, otherwise, it will ask the same question again.

A typical example of the display of the program (once a user chooses the option **<Add>**) can be as follows.

```
= A
Please enter the student ID: 123
Please enter the student name: abc
Please enter the course: c101
Thank You!
The details of the student you entered are as follows:
Student ID: 123
Student name: abc
Please enter the course: c101
The record has been successfully added to the students.txt file.
Do you want to enter Details for another student (Y/N)?
= Y
Please enter the student ID: 234
Please enter the student name: def
Please enter the course: c102
Thank You!
The details of the student you entered are as follows:
Student ID: 234
Student name: def
Please enter the course: c102
The record has been successfully added to the students.txt file.
Do you want to enter Details for another student (Y/N)?
= Y
Please enter the student ID: 344
Please enter the student name: ghi
Please enter the course: c101
Thank You!
The details of the student you entered are as follows:
Student ID: 344
Student name: ghi
Please enter the course: c101
The record has been successfully added to the students.txt file.
Do you want to enter Details for another student (Y/N)?
= N
```

After the operation, the **students.txt** file will have the following content:



2. Insert Option

In a case where the main menu input by user is **<Insert>**, then the user must enter the student ID and subject code, assessment number and marks in the same order. After receiving all the inputs from the user, details will be displayed on the console.

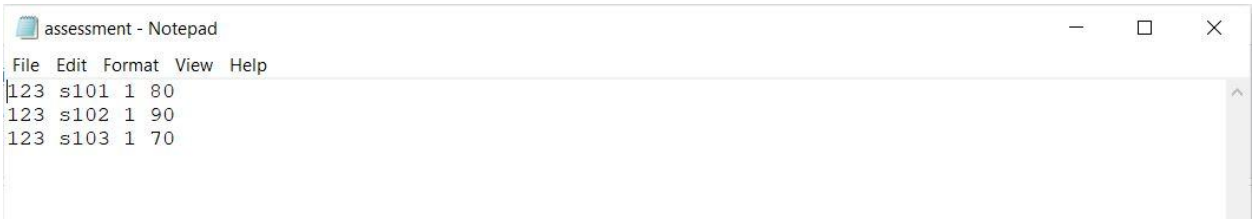
Subsequently the details of the assessment marks should be added in the **assessments.txt** file. Here you have to again ensure that the student data exists in the file and he/she is a valid student.

Once the new record has been added, the system will then ask the user **'Do you want to enter marks for another assessment (Y/N)?'**

- If they enter **'Y'** the system will allow them to enter marks for another assessment as before.
- If they enter **'N'** the system will display the main menu again, otherwise, it will ask the same question again.

A typical example of the display of the program (once a user chooses the option **<Insert>**) can be as follows. Your program can follow a different display style. Just make sure that the behavior is similar to what has been mentioned.

```
======================================================================
= I
Please enter the student ID: 123
Please enter the subject code: s101
Please enter assessment No.: 1
Please enter assessment Marks: 80
Thank You!
The details of the student you entered are as follows:
Student ID: 123
Subject Code: s101
Assessment No 1
Assessment Marks 80
The record has been successfully added to the assessments.txt file.
Do you want to enter marks for another assessment (Y/N)?
= Y
Please enter the subject code: s102
Please enter assessment No.: 1
Please enter assessment Marks: 90
Thank You!
The details of the student you entered are as follows:
Student ID: 123
Subject Code: s102
Assessment No 1
Assessment Marks 90
The record has been successfully added to the assessments.txt file.
Do you want to enter marks for another assessment (Y/N)?
= Y
Please enter the subject code: s103
Please enter assessment No.: 1
Please enter assessment Marks: 70
Thank You!
The details of the student you entered are as follows:
Student ID: 123
Subject Code: s103
Assessment No 1
Assessment Marks 70
The record has been successfully added to the assessments.txt file.
Do you want to enter marks for another assessment (Y/N)?
= N
======================================================================
```



3. Search option

Once the user asks for <**Search**> from the main menu then the user will be asked to enter the student ID for whom they want to see details.

The program then collects the student details from the **students.txt** file and assessment marks for that student from the **assessments.txt** file and displays them as follows (assuming the following student was searched for).

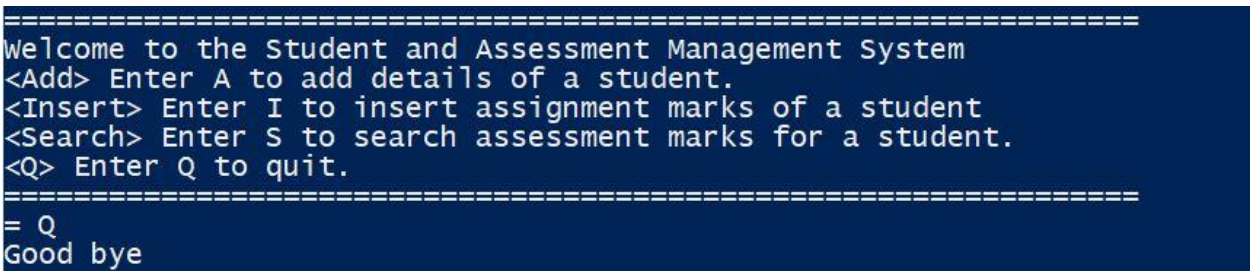
```
= S
Please enter the student ID you want to search assessment marks for: 123
Thank You
A student has been found:
Student ID:  123
Student Name:  abc
Course:  c101
Subject Code    Assessment Number       Marks
s101            1                       80

s102            1                       90

s103            1                       70

Do you want to search assessment marks for another student (Y/N)?
=
```

After displaying the assessment marks the program prompts the user with the following message, **'Do you want to search assessment marks for another student (Y/N)?'**

- If a user enters **'Y'** then the program asks them to enter the student ID for which the assessment marks need to be searched and displayed.
- If the user enters **'N'** then the program displays the main menu, otherwise, the program prompts the same message again.

4. Quit option

Finally, the program quits if the user chooses the option **<Q>.**



Your program should be able to handle invalid inputs such as not-a-number or negative numbers.

In addition, for this exercise, you can follow the following basic rules:

I. Use multiple functions, instead of using a single function to do everything.
II. Create a good design of the functions to make the best use of the code and avoid duplicate calculations.
III. Avoid duplicate code.
IV. You also need to design your program so that it has components that can be reused in another program if needed.
V. Handle exceptions appropriately.
VI. Use appropriate data structure.