

Mastering Embedded System Online Diploma

www.learn-in-depth.com



First Term Project 2

Student Database System



BY :
Eng. Mohamed Kamel Aly

My Profile : No progress page created,
Email: mohamed.kamel.m73@gmail.com

Contents

Problem Statement	3
Approach	3
Configuration	3
Code APIs And Helper Methods.....	4
APIs.....	4
addManually	4
addFromFile	4
printStudentByID.....	4
printByFirstName	4
printCourseStudents	4
printAll.....	4
deleteByID	4
updateByID.....	4
countStudents	5
HelperMethods'	5
isUniqueID	5
getStudentByID	5
printStudentInfo.....	5
FIFO_validator	5
enterStudentInfo.....	5
Tests	5
Add students from file:	6
Print All Students	7
Add Student Manually	7
Print Student by ID	8
Print Student By First Name	8
Print Students In Course	9
Delete Student By ID	10
Update Student By ID.....	10
Get Total Number Of Students.....	11

Problem Statement

Write a program to build a simple Software for Student Information Management System which can perform the following operations:

- Store the First name of the student.
- Store the Last name of the student.
- Store the unique Roll number for every student.
- Store the GPA of every student.
- Store the courses registered by the student.

Approach

The idea is to form an individual functions for every operation. All the functions are unified together to form software.

- Add Student Details From File
- Add Student Details Manually
- Find the student by the given roll number
- Find the student by the given first name
- Find the students registered in a course
- Count of students
- Delete a student
- Update Student

Configuration

In students.h, there are some config options to se

```
// ////////////////////////////////////////  
// Configuration  
// ////////////////////////////////////////  
#define BUFFER_LENGTH 50  
#define MAX_NUM_OF_COURSES 10  
#define MAX_NAME_CHARACTERS 50  
#define FILENAME "studentsDB.txt"  
BUFFER_LENGTH:
```

Sets the maximum number of students to save in the DB

MAX_NUM_OF_COURSES:

Sets the maximum No. of courses each single student can take

MAX_NAME_CHARACTERS:

Sets the maximum first/last name No. of characters

Code APIs And Helper Methods

Github Code:

APIs

These are the main APIs used to provide functionality

addManually

```
void addManually(FIFO_Buf_t *S_FIFO);
```

This API adds a student manually, it uses helper methods of **FIFO_validator**, **enterStudentInfo**, **countStudents**

addFromFile

```
void addFromFile(FIFO_Buf_t *S_FIFO);
```

This API adds students from an existing text file, it uses helper methods of **FIFO_validator**, **isUniqueID**

printStudentByID

```
void printStudentByID(FIFO_Buf_t *S_FIFO);
```

This API prints a student by ID, it uses helper methods of **FIFO_validator**, **getStudentByID**, **printStudentInfo**

printByFirstName

```
void printByFirstName(FIFO_Buf_t *S_FIFO);
```

This API prints a student by ID, it uses helper methods of **FIFO_validator**, **printStudentInfo**

printCourseStudents

```
void printCourseStudents(FIFO_Buf_t *S_FIFO);
```

This API prints students taking a certain course ID, it uses helper methods of **FIFO_validator**, **printStudentInfo**

printAll

```
void printAll(FIFO_Buf_t *S_FIFO);
```

This API prints All students, it uses helper methods of **FIFO_validator**, **printStudentInfo**

deleteByID

```
void deleteByID(FIFO_Buf_t *S_FIFO);
```

This API deletes a student by ID, it uses helper methods of **FIFO_validator**, **printStudentInfo**, **getStudentByID**

updateByID

```
void updateByID(FIFO_Buf_t *S_FIFO);
```

This API updates a student by ID, it uses helper methods of **FIFO_validator**, **getStudentByID**, **enterStudentInfo**

countStudents

```
void countStudents(FIFO_Buf_t *S_FIFO);
```

This API counts all students, it uses the helper method **FIFO_validator**

HelperMethods

These are the helper methods used by the main APIs to enhance code readability/maintainability

isUniqueID

```
bool isUniqueID(FIFO_Buf_t *S_FIFO, uint32 ID);
```

This method is called to check if a given ID input is unique or not, it uses the helper method

getStudentByID

getStudentByID

```
Student *getStudentByID(FIFO_Buf_t *S_FIFO, uint32 ID, int  
*studentListingNumP);
```

This method iterates on student IDs to return a pointer to the student with the ID, if not found, it returns NULL, it uses the helper method **FIFO_validator**

printStudentInfo

```
void printStudentInfo(Student *s);
```

This method prints a given student's information

FIFO_validator

```
bool FIFO_validator(FIFO_Buf_t *S_FIFO, bool isEmptyValidate, bool  
isFullValidate);
```

This method is used to validate the FIFO buffer and its' data to check if it's invalid, empty, or full

enterStudentInfo

```
void enterStudentInfo(FIFO_Buf_t *S_FIFO, Student *targetStudent, uint8  
choice);
```

This method is used to enter student data and save it the student pointer

Tests

These are the options which are presented to the user upon starting the program

```
Choose One Of The Following Options  
1: Add Student(Manually)  
2: Add Student(Text File)  
3: Print All Students  
4: Print Student(By ID)  
5: Print Student(By First Name)  
6: Print All Students Enrolled In A Course(Course ID)  
7: Delete Student(By ID)  
8: Update Student(By ID)  
9: Get Total Number Of Students  
10: Exit The Program  
3
```

Add students from file:

While reading each line, any non-unique ID entry is skipped and the program continues saving from the next line

```
1
2 2 mohamed kamel 2.00 5 2 5 2 3 2
3 3 ahmed ali 1.20 5 2 5 2 3 2
4 54 ahmaed ali 1.20 3 23 64 35
5 34 ahmed kheded 3.40 5 23 64 333 33 0
6 14 ahmed kamek 2.00 3 33 44 50
7 23 mohamed kamel 2.30 2 23 23
8 132 mohaededd edas 2.30 2 2 5
9 54 mohamed kamel 3.20 1 2
10 23 mohamed kamel 3.20 1 2
11 13 mohamed kamel 3.03 2 23 23

<

Console x Regist... Proble... Execu... Debug...
PlayGround.exe [C/C++ Application] [pid: 3]
5: Print All Students
4: Print Student(By ID)
5: Print Student(By First Name)
6: Print All Students Enrolled In A Course(Course
7: Delete Student(By ID)
8: Update Student(By ID)
9: Get Total Number Of Students
10: Exit The Program
2
=====
* Student with ID 2 Saved Successfully
* Student with ID 3 Saved Successfully
* Student with ID 54 Saved Successfully
* Student with ID 34 Saved Successfully
* Student with ID 14 Saved Successfully
* Student with ID 23 Saved Successfully
* Student with ID 132 Saved Successfully
=====
* ID 54 is Already Taken
=====

=====
* ID 23 is Already Taken
=====

* Student with ID 13 Saved Successfully
=====
```

Print All Students

The Students added from the previous file are present when choosing option 3 (PrintAll)

```
=====
=====Student=====
=====+=====
ID:      2
First Name is:  mohamed
Last Name is:   kamel
GPA:      2.00
No. Of Courses Enrolled:      5
Course Number 1 ID is 2
Course Number 2 ID is 5
Course Number 3 ID is 2
Course Number 4 ID is 3
Course Number 5 ID is 2
=====
=====Student=====
=====+=====
ID:      3
First Name is:  ahmed
Last Name is:   ali
GPA:      1.20
No. Of Courses Enrolled:      5
Course Number 1 ID is 2
Course Number 2 ID is 5
Course Number 3 ID is 2
Course Number 4 ID is 3
Course Number 5 ID is 2
=====
=====Student=====
=====+=====
ID:      54
First Name is:  ahmaed
Last Name is:   ali
GPA:      1.20
No. Of Courses Enrolled:      3
Course Number 1 ID is 23
Course Number 2 ID is 64
=====
```

Add Student Manually

The left screenshot show the process of adding the student

, and the right one shows it's been added by printing all students

```
9: Get Total Number Of Students
10: Exit The Program
1
=====
Enter Student First Name : Ahmed
Enter Student Last Name : Khaled
Enter A Unique Student ID : 56
Enter Student GPA : 2.3
Enter The Number Of Courses choosen (MAX = 10): 3
Enter ID for Course No. 1 : 34
Enter ID for Course No. 2 : 35
Enter ID for Course No. 3 : 22
Student added successfully
=====The Total Number Of Students is 9
You Can Add Up To 50 Students
You Can Add 41 More Students
=====
```

```
=====
No. Of Courses Enrolled:      2
Course Number 1 ID is 23
Course Number 2 ID is 23
=====
=====Student=====
=====+=====
ID:      56
First Name is:  Ahmed
Last Name is:   Khaled
GPA:      2.30
No. Of Courses Enrolled:      3
Course Number 1 ID is 34
Course Number 2 ID is 35
Course Number 3 ID is 22
=====
```

Print Student by ID

From the above added students, searching for a student with an ID prints the target correctly

```
10: Exit The Program
4
=====
Enter Student ID:23
=====Student=====
=====
ID:      23
First Name is: mohamed
Last Name is:  kamel
GPA:     2.30
No. Of Courses Enrolled:      2
Course Number 1 ID is 23
Course Number 2 ID is 23
=====
```

Non-existing IDs print the following

```
10: Exit The Program
4
=====
Enter Student ID:234
=====
Student ID not found=====
Choose One Of The Following Options
```

Print Student By First Name

This also works by entering the name, then the found student is printed, entering a non-existing name is shown in the right screenshot

```
10: Exit The Program
5
=====
Enter First name to search:ahmed
=====Student=====
=====
ID:      3
First Name is: ahmed
Last Name is:  ali
GPA:     1.20
No. Of Courses Enrolled:      5
Course Number 1 ID is 2
Course Number 2 ID is 5
Course Number 3 ID is 2
Course Number 4 ID is 3
Course Number 5 ID is 2
=====
```

```
10: Exit The Program
5
=====
Enter First name to search:ksk
=====
Student name not found=====
Choose One Of The Following Options
```


Print Students In Course

The students taking course No 33 are all printed as intended

Entering a non-existing course will print the following message:

```
10: Exit The Program
6
=====
Enter Course ID to search:343
=====
No Students found taking such course=====
```

```
6
=====
Enter Course ID to search:33
=====Student=====
=====
ID:      34
First Name is:  ahmed
Last Name is:   kheded
GPA:      3.40
No. Of Courses Enrolled:      5
Course Number 1 ID is 23
Course Number 2 ID is 64
Course Number 3 ID is 333
Course Number 4 ID is 33
Course Number 5 ID is 0
=====
=====Student=====
=====
ID:      14
First Name is:  ahmed
Last Name is:   kamek
GPA:      2.00
No. Of Courses Enrolled:      3
Course Number 1 ID is 33
Course Number 2 ID is 44
Course Number 3 ID is 50
=====
```

Delete Student By ID

In this test, we delete the student with the ID 132, the lower screenshots show the DB before and after the deletion

```
7
=====
Enter Student ID to delete132
=====
```

```
=====Student=====
ID: 23
First Name is: mohamed
Last Name is: kamel
GPA: 2.30
No. Of Courses Enrolled: 2
Course Number 1 ID is 23
Course Number 2 ID is 23
=====Student=====
ID: 132
First Name is: mohaededd
Last Name is: edas
GPA: 2.30
No. Of Courses Enrolled: 2
Course Number 1 ID is 2
Course Number 2 ID is 5
=====Student=====
ID: 13
First Name is: mohamed
Last Name is: kamel
GPA: 3.03
No. Of Courses Enrolled: 2
Course Number 1 ID is 23
Course Number 2 ID is 23
=====
```

```
=====Student=====
ID: 23
First Name is: mohamed
Last Name is: kamel
GPA: 2.30
No. Of Courses Enrolled: 2
Course Number 1 ID is 23
Course Number 2 ID is 23
=====Student=====
ID: 13
First Name is: mohamed
Last Name is: kamel
GPA: 3.03
No. Of Courses Enrolled: 2
Course Number 1 ID is 23
Course Number 2 ID is 23
=====Student=====
ID: 56
```

Update Student By ID

The left-most Screen shows a student we will update, upon entering student ID, we choose what we want to update, the right-most screen shows that the student has been updated successfully

```
ID: 56
First Name is: Ahmed
Last Name is: Khaled
GPA: 2.30
No. Of Courses Enrolled: 3
Course Number 1 ID is 34
Course Number 2 ID is 35
Course Number 3 ID is 22
=====
```

```
8: Update Student(By ID)
9: Get Total Number Of Students
10: Exit The Program
8
=====
Enter Student ID to update56
1) Update First Name
2) Update Last Name
3) Update ID
4) Update GPA
5) Update Courses
4
Enter Student GPA : 3.1
=====
Student Updated successfully=====
```

```
=====Student=====
Enter Student ID:56
=====Student=====
ID: 56
First Name is: Ahmed
Last Name is: Khaled
GPA: 3.10
No. Of Courses Enrolled: 3
Course Number 1 ID is 34
Course Number 2 ID is 35
Course Number 3 ID is 22
=====
```

Get Total Number Of Students

This functionality shows the number of students in our database

```
9: Get Total Number Of Students
10: Exit The Program
9
=====
The Total Number Of Students is 8
You Can Add Up To 50 Students
You Can Add 42 More Students
```