

Nov 2021

First Term (Final Project 2) Student Management System

Eng. Mazen Talaat

My profile: <https://bit.ly/3DUmnyz>

Mastering embedded system online diploma by Eng. Keroles Shenouda
www.learn-in-depth.com

Table of contents

Problem Statement	2
Approach.....	2
Idea.....	2
main.c.....	3
SYS_API.h.....	4
SYS_API.c	5
1. Add student details manually	5
2. Add student details from file.....	6
3. Find the student by the given roll number	7
4. Find the student by the given first name	7
5. Find the student registered in a course	8
6. Count number of students	8
7. Delete a student.....	9
8. Update a student.....	10
9. Print data	11
Students.txt.....	11
Implemented code output.....	12

Problem Statement

A simple software for student information management system which can perform the following operations:

1. Store first name of the student.
2. Store last name of the student.
3. Store unique roll number for every student.
4. Store GPA for every student.
5. Store courses registered by the student.

Approach

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

1. Add student details from file.
2. Add student details manually.
3. Find the student by the given roll number.
4. Find the student by the given first name.
5. Find the student registered in a course.
6. Count number of students.
7. Delete a student by the given roll number.
8. Update a student by the given roll number.
9. Print all student's data.
10. Exit the program.

Idea

The software will consist of 4 files main.c, SYS_API.c, SYS_API.h and, Students.txt.

The main.c will contain the interface and calling the function, SYS_API.c will contain the global variables and the body of the functions, SYS_API.h will contain the definitions and the prototypes, Students.txt will contain the data.

1. We will have array of struct of 50 elements which is the max number of students.
2. Each struct contains the student details.
3. We will have a global index which refers to the number of students.
4. The index is initialized with zero and can go up to 50.

main.c

The main.c is just a simple file which consists of:

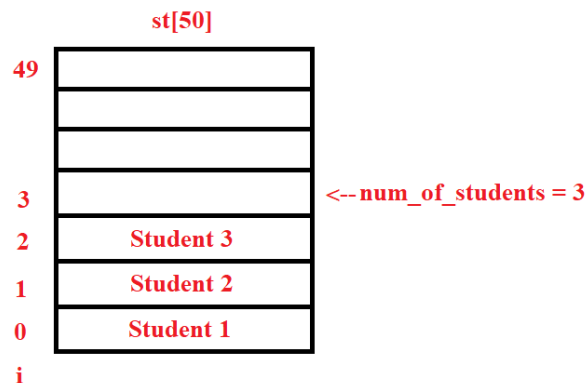
- Infinite while loop until the choice = 10.
- Dprint and Dscan function(a macro to printf and fflush to fix eclipse console bug).
- Calling the program functions through switch cases.

```
8      #include "SYS_API.h"
9
10     int main(){
11         int choice;
12         Dprint(" Welcome to the student management system\n");
13         while(1){
14             Dprint("\n Choose the task that you want to perform\n");
15             Dprint(" 1. Add a student details manually\n");
16             Dprint(" 2. Add a student details from a text file\n");
17             Dprint(" 3. Find student details by roll number\n");
18             Dprint(" 4. Find student details by first name\n");
19             Dprint(" 5. Find student details by course ID\n");
20             Dprint(" 6. Find the total number of students\n");
21             Dprint(" 7. Delete student details by roll number\n");
22             Dprint(" 8. Update student details by roll number\n");
23             Dprint(" 9. Show all information\n");
24             Dprint(" 10. Exit\n");
25             Dprint(" Enter your choice to perform the task: ");
26             Dscan("%d", &choice);
27             switch(choice){
28                 case 1:
29                     add_student_manually();
30                     break;
31                 case 2:
32                     add_student_file();
33                     break;
34                 case 3:
35                     find_rl();
36                     break;
37                 case 4:
38                     find_fn();
39                     break;
40                 case 5:
41                     find_c();
42                     break;
43                 case 6:
44                     tot_s();
45                     break;
46                 case 7:
47                     del_s();
48                     break;
49                 case 8:
50                     up_s();
51                     break;
52                 case 9:
53                     show_s();
54                     break;
55                 case 10:
56                     Dprint(" Bye!\n");
57                     return 1;
58                 default:
59                     Dprint(" Wrong input value\n");
60                     break;
61             }
62         }
63         return 0;
64     }
```

SYS_API.h

The SYS_API.h consist of:

- Macros to make things easier.
- Struct sinfo which contains the details of a student.
- st[max_num_of_students] which is an array of 50 of struct sinfo.
- FILE *fp which is a pointer to a typedef FILE to manage reading a file.
- Functions prototypes with their usage.



```

8  #ifndef SYS_API_H_
9  #define SYS_API_H_
10
11  #include <stdio.h>
12  #include <stdlib.h>
13  #include <string.h>
14
15  /* Macros */
16  #define Dprint(...) printf(__VA_ARGS__); fflush(stdin); fflush(stdout)
17  #define Dscan(...) scanf(__VA_ARGS__); fflush(stdin); fflush(stdout)
18
19  #define name_length 50
20  #define num_of_courses 5
21  #define max_num_of_students 50
22
23  /* Definitions */
24  struct sinfo{
25      char  fname[name_length];
26      char  lname[name_length];
27      int   roll;
28      float gpa;
29      int   cid[num_of_courses];
30  }st[max_num_of_students];
31
32  FILE *fp;
33
34  /* APIs */
35  void add_student_manually(    ); /* Add student details manually          */
36  void add_student_file      (    ); /* Add student details from text file */
37  void find_rl               (    ); /* Find the student by the given roll number */
38  void find_fn               (    ); /* Find the student by the given first name */
39  void find_c                (    ); /* Find the student registered in a course */
40  void tot_s                 (    ); /* Count number of students            */
41  void remove_gap            (int); /* Fills the gap                      */
42  void del_s                 (    ); /* Delete a student                   */
43  void up_s                  (    ); /* Update a student                   */
44  void show_s                (    ); /* Print student details              */
45
46  #endif /* SYS_API_H_ */
47

```

SYS_API.c

The SYS_API.h consist of:

- Global variable num_of_students which is the index of the array st[] to track the data.
- Functions body.

1. Add student details manually

- The function will start by checking the list if it is full.
- Takes from user the roll number and checks if it is duplicated.
 - a. If yes, it will print an error.
 - b. If no, it will print success and add the student details to the location where the index is pointing to.
- The number of students is then incremented by 1.
- Printing the total, max and remaining number of students through the function tot_s().

```
8  #include "SYS_API.h"
9
10 /* Global variable */
11 int num_of_students = 0;
12
13 /* Functions body */
14 /* Add student details manually */
15 void add_student_manually(){
16     /* Check if full */
17     if(num_of_students < 50){
18         int i, rn;
19         Dprint(" -----\\n");
20         Dprint(" Add the student details\\n");
21         Dprint(" -----\\n");
22         Dprint(" Enter the roll number: ");
23         Dscan("%d", &rn);
24         for(i=0; i<num_of_students; i++){
25             if(st[i].roll == rn){
26                 Dprint(" [ERROR] Roll number %d is already taken\\n", rn);
27                 return;
28             }
29         }
30         st[num_of_students].roll = rn;
31         Dprint(" Enter first name of student: ");
32         Dscan("%s", st[num_of_students].fname);
33         Dprint(" Enter last name of student: ");
34         Dscan("%s", st[num_of_students].lname);
35         Dprint(" Enter the GPA you obtained: ");
36         Dscan("%f", &st[num_of_students].gpa);
37         Dprint(" Enter the courses ID\\n");
38         for(i = 0; i<5; i++){
39             Dprint(" Course %d ID: ", i+1);
40             Dscan("%d", &st[num_of_students].cid[i]);
41         }
42         num_of_students++;
43         Dprint(" [INFO] Students details is added successfully\\n");
44         tot_s();
45     }
46     else{
47         Dprint(" [ERROR] The list is full\\n");
48     }
49 }
```

2. Add student details from file

- The function will start by checking the list if it is full.
- Reads from the file the details, saves them, and checks if the roll number is duplicated.
 - a. If yes, it will print that the roll number is already taken and continue to the next line in the file.
 - b. If no, it will print success and increment the number of students by 1 (index).
- If it reached the EOF(end of file) , a flag will be raised.
- Printing the total, max and remaining number of students through the function tot_s().

```
51  /* Add student details from text file */
52  void add_student_file(){
53      /* Check if full */
54      if(num_of_students < 50){
55          /*Buffer to store the data from file */
56          char buff[100];
57          int i, rn, F_found, F_EOF;
58          fp = fopen("Students.txt", "r");
59          while(1){
60              /* Reset the flag */
61              F_found = 0;
62              /* Read Roll number */
63              fscanf(fp, "%s", buff);
64              rn = atoi(buff);
65              st[num_of_students].roll = rn;
66              /* Read First name */
67              fscanf(fp, "%s", st[num_of_students].fname);
68              /* Read Last name */
69              fscanf(fp, "%s", st[num_of_students].lname);
70              /* Read GPA */
71              fscanf(fp, "%s", buff);
72              st[num_of_students].gpa = atof(buff);
73              for(i=0; i<num_of_courses; i++){
74                  /* Read Courses ID */
75                  F_EOF = fscanf(fp, "%s", buff);
76                  st[num_of_students].cid[i] = atoi(buff);
77              }
78              /* Search for roll ID */
79              for(i=0; i<num_of_students; i++){
80                  if(st[i].roll == rn){
81                      Dprint(" [ERROR] Roll number %d is already taken\n", rn);
82                      /* Raise found flag */
83                      F_found = 1;
84                      break;
85                  }
86              }
87              if(F_found == 0){
88                  Dprint(" [INFO] Roll number %d saved successfully\n", rn);
89                  num_of_students++;
90              }
91              /* Check if reached EOF */
92              if(F_EOF == -1){
93                  fclose(fp);
94                  break;
95              }
96          }
97          Dprint(" [INFO] Students details is added successfully\n");
98          tot_s();
99      }
100  else{
101      Dprint(" [ERROR] The list is full\n");
102  }
103 }
```

3. Find the student by the given roll number

- The function will start by checking the list if it is empty.
- Searches for the roll number received from user.
 - a. If found, it will print that the details and return.
 - b. If not, it will print an error.

```

105  /* Find the student by the given roll number */
106  void find_rl(){
107      /* Check if empty */
108      if(num_of_students != 0){
109          int i, j, rn;
110          Dprint("-----\n");
111          Dprint(" Find a student by roll number\n");
112          Dprint("-----\n");
113          Dprint(" Enter the roll number: ");
114          Dscan("%d", &rn);
115          /* Search for roll number */
116          for(i = 0; i<num_of_students; i++){
117              if (st[i].roll == rn){
118                  Dprint("-----\n");
119                  Dprint("\n Student first name: %s\n", st[i].fname);
120                  Dprint(" Student last name: %s\n", st[i].lname);
121                  Dprint(" Student roll number: %d\n", st[i].roll);
122                  Dprint(" Student GPA number: %.1f\n", st[i].gpa);
123                  for(j=0; j<num_of_courses; j++){
124                      Dprint(" Course %d ID: %d\n", j+1, st[i].cid[j]);
125                  }
126                  Dprint("-----\n");
127                  return;
128              }
129          }
130          /* If the loop ended */
131          Dprint(" [ERROR] Roll number %d not found\n", rn);
132      }
133      else{
134          Dprint(" [ERROR] The list is empty!\n");
135      }
136  }

```

4. Find the student by the given first name

- The function will start by checking the list if it is empty.
- Searches for the first name received from user.
 - a. If found, it will print that the details and raise a flag.
 - b. If not, it will print an error.
- It will continue the search until reaching the end of the list.

```

138  /* Find the student by the given first name */
139  void find_fn(){
140      /* Check if empty */
141      if(num_of_students != 0){
142          int i, j, F_found=0;
143          char fn[name_length];
144          Dprint("-----\n");
145          Dprint(" Find a student by first name\n");
146          Dprint("-----\n");
147          Dprint(" Enter the first name: ");
148          Dscan("%s", fn);
149          /* Search for first name by comparing strings */
150          for(i = 0; i<num_of_students; i++){
151              if (!strcmp(fn, st[i].fname)){
152                  Dprint("-----\n");
153                  Dprint(" Student first name: %s\n", st[i].fname);
154                  Dprint(" Student last name: %s\n", st[i].lname);
155                  Dprint(" Student roll number: %d\n", st[i].roll);
156                  Dprint(" Student GPA number: %.1f\n", st[i].gpa);
157                  for(j=0; j<num_of_courses; j++){
158                      Dprint(" Course %d ID: %d\n", j+1, st[i].cid[j]);
159                  }
160                  /* Raise found flag */
161                  F_found = 1;
162              }
163          }
164          if (F_found == 0)
165              Dprint(" [ERROR] First name %s not found\n", fn);
166      }
167      else{
168          Dprint(" [ERROR] The list is empty!\n");
169      }
170  }

```


5. Find the student registered in a course

- The function will start by checking the list if it is empty.
- Searches for the course ID received from user.
 - a. If found, it will print that the details and raise a flag.
 - b. If not, it will print an error.
- It will continue the search until reaching the end of the list and prints the total number of students enrolled.

```
172  /* Find the student registered in a course */
173  void find_c(){
174      /* Check if empty */
175      if(num_of_students != 0){
176          int i, j, ID, F_found=0;
177          int count = 0;
178          Dprint(" -----\\n");
179          Dprint(" Find students by course ID\\n");
180          Dprint(" -----\\n");
181          Dprint(" Enter the course ID: ");
182          Dscan("%d", &ID);
183          /* Search for course ID */
184          for(i = 0; i<num_of_students; i++){
185              for(j=0; j<num_of_courses; j++){
186                  if (st[i].cid[j] == ID){
187                      Dprint(" -----\\n");
188                      Dprint(" Student first name: %s\\n", st[i].fname);
189                      Dprint(" Student last name: %s\\n", st[i].lname);
190                      Dprint(" Student roll number: %d\\n", st[i].roll);
191                      Dprint(" Student GPA number: %.1f\\n", st[i].gpa);
192                      /* Increase the number of enrolled students by 1 */
193                      count++;
194                      /* Raise found flag */
195                      F_found = 1;
196                      /* Break to stop the courses loop */
197                      break;
198                  }
199              }
200          }
201          if (F_found == 0){
202              Dprint(" [ERROR] Course ID %d not found\\n", ID);
203          }
204          else{
205              Dprint(" [INFO] Total number of students enrolled: %d\\n", count);
206          }
207      }
208      else{
209          Dprint(" [ERROR] The list is empty!\\n");
210      }
211  }
```

6. Count number of students

- The function will print the total number of students by using the index_number_of_students.
- Print the max number by using the macro max_number_of_students.

```
213  /* Count number of students */
214  void tot_s(){
215      Dprint(" -----\\n");
216      Dprint(" [INFO] The total number of students is %d\\n", num_of_students);
217      Dprint(" [INFO] You can add up to %d students\\n", max_num_of_students);
218      Dprint(" [INFO] You can add %d more students\\n", max_num_of_students - num_of_students);
219  }
```

7. Delete a student

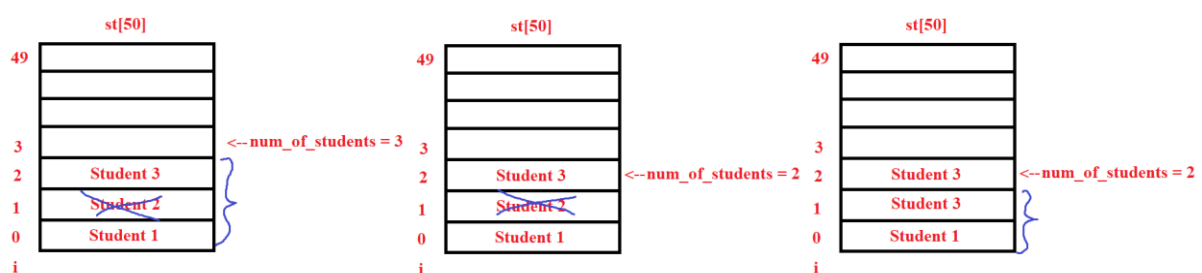
It consists of two functions: `remove_gap` and `del_s`.

`del_s`:

- The function will start by checking the list if it is empty.
- Searches for the roll number received from user.
 - a. If found, it will decrease the number of students, calls the function `remove_gap` and return.
 - b. If not, it will print an error.

`remove_gap`:

- The function will shift the elements starting from the index of the removed student to the start.



```

221  /* Fills the gap */
222  void remove_gap(int index){
223      int i;
224      for(i=index; i<num_of_students; i++){
225          /* Shift all students to the removed index */
226          st[i] = st[i+1];
227      }
228  }
229
230  /* Delete a student */
231  void del_s(){
232      /* Check if empty */
233      if(num_of_students != 0){
234          int i, rn;
235          Dprint(" ----- \n");
236          Dprint(" Delete a student by roll number \n");
237          Dprint(" ----- \n");
238          Dprint(" Enter the roll number: ");
239          Dscan("%d", &rn);
240          /* Search for roll number */
241          for(i = 0; i<num_of_students; i++){
242              if (st[i].roll == rn){
243                  /* Decrease the num_of_students by 1 */
244                  num_of_students--;
245                  /* Fill the gap with the next student */
246                  remove_gap(i);
247                  Dprint(" [INFO] The roll number is removed successfully \n");
248                  return;
249              }
250          }
251          /* If the loop ended */
252          Dprint(" [ERROR] The roll number %d not found \n", rn);
253      }
254      else{
255          Dprint(" [ERROR] The list is empty! \n");
256      }
257  }
258  }

```

8. Update a student

- The function will start by checking the list if it is empty.
- Searches for the roll number received from user.
 - a. If found, it will ask the user to choose what to update and print success if so.
 - b. If not, it will print an error.

```
260  /* Update a student */
261  void up_s(){
262      /* Check if empty */
263      if(num_of_students != 0){
264          int i, j, rn, x;
265          Dprint(" ----- \n");
266          Dprint(" Update a student by roll number \n");
267          Dprint(" ----- \n");
268          Dprint(" Enter the roll number: ");
269          Dscan("%d", &rn);
270          /* Search for roll number */
271          for(i = 0; i < num_of_students; i++){
272              if (st[i].roll == rn){
273                  Dprint(" 1. First name \n");
274                  Dprint(" 2. Last name \n");
275                  Dprint(" 3. Roll number \n");
276                  Dprint(" 4. GPA \n");
277                  Dprint(" 5. Courses \n");
278                  Dprint(" Enter your choice: ");
279                  Dscan("%d", &x);
280                  switch(x){
281                      case 1:
282                          Dprint(" Enter the new first name: ");
283                          Dscan("%s", st[i].fname);
284                          Dprint(" [INFO] Updated successfully \n");
285                          break;
286                      case 2:
287                          Dprint(" Enter the new last name: ");
288                          Dscan("%s", st[i].lname);
289                          Dprint(" [INFO] Updated successfully \n");
290                          break;
291                      case 3:
292                          Dprint(" Enter the new roll number: ");
293                          Dscan("%d", &st[i].roll);
294                          Dprint(" [INFO] Updated successfully \n");
295                          break;
296                      case 4:
297                          Dprint(" Enter the new GPA: ");
298                          Dscan("%f", &st[i].gpa);
299                          Dprint(" [INFO] Updated successfully \n");
300                          break;
301                      case 5:
302                          Dprint(" Enter the new course ID: \n");
303                          for(j = 0; j < 5; j++){
304                              Dprint(" Course %d ID: ", j+1);
305                              Dscan("%d", &st[i].cid[j]);
306                          }
307                          Dprint(" [INFO] Updated successfully \n");
308                          break;
309                      default:
310                          Dprint(" Wrong input value \n");
311                          break;
312                  }
313                  return;
314              }
315          }
316          /* If the loop ended */
317          Dprint(" [ERROR] Roll number %d not found \n", rn);
318      }
319      else{
320          Dprint(" [ERROR] The list is empty! \n");
321      }
322  }
```

9. Print data

- The function will start by checking the list if it is empty.
- Loop through each element and print the data.

```
324 /* Print student details */
325 void show_s(){
326     /* Check if empty */
327     if(num_of_students != 0){
328         int i, j;
329         Dprint(" -----\\n");
330         for(i=0; i<num_of_students; i++){
331             Dprint("\\n Student first name: %s\\n", st[i].fname);
332             Dprint(" Student last name: %s\\n" , st[i].lname);
333             Dprint(" Student roll number: %d\\n" , st[i].roll);
334             Dprint(" Student GPA number: %.1f\\n", st[i].gpa);
335             for(j=0; j<num_of_courses; j++){
336                 Dprint(" Course %d ID: %d\\n", j+1, st[i].cid[j]);
337             }
338         }
339         Dprint(" -----\\n");
340     }
341     else{
342         Dprint("[ERROR] The list is empty!\\n");
343     }
344 }
345
```

Students.txt

The format is:

Roll_Number First_Name Last_Name GPA C1_ID C2_ID C3_ID C4_ID C5_ID

1	1	Marco	Magdy	3.5	1	2	3	4	5
2	1	Pavly	Salah	3	80	12	37	29	1
3	2	Kerolos	Shenouda	3.9	99	1	55	66	77
4	3	Bolis	Karam	3.5	45	21	55	18	46
5	4	Kerolos	Gamal	3.5	45	21	55	18	

Implemented code output

```
Welcome to the student management system

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 1
-----
Add the student details
-----
Enter the roll number: 5
Enter first name of student: Mazen
Enter last name of student: Talaat
Enter the GPA you obtained: 4
Enter the courses ID
Course 1 ID: 9
Course 2 ID: 8
Course 3 ID: 7
Course 4 ID: 6
Course 5 ID: 5
[INFO] Student details is added successfully
-----
[INFO] The total number of students is 1
[INFO] You can add up to 50 students
[INFO] You can add 49 more students

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 9
-----

Student first name: Mazen
Student last name: Talaat
Student roll number: 5
Student GPA number: 4.0
Course 1 ID: 9
Course 2 ID: 8
Course 3 ID: 7
Course 4 ID: 6
Course 5 ID: 5
-----
```

```

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 2
[INFO] Roll number 1 saved successfully
[ERROR] Roll number 1 is already taken
[INFO] Roll number 2 saved successfully
[INFO] Roll number 3 saved successfully
[INFO] Roll number 4 saved successfully
[INFO] Students details is added successfully
-----
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 6
-----
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 3
-----
Find a student by roll number
-----
Enter the roll number: 5
-----

Student first name: Mazen
Student last name: Talaat
Student roll number: 5
Student GPA number: 4.0
Course 1 ID: 9
Course 2 ID: 8
Course 3 ID: 7
Course 4 ID: 6
Course 5 ID: 5
-----

```

```
Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 3
-----
Find a student by roll number
-----
Enter the roll number: 55
[ERROR] Roll number 55 not found
```

```
Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 4
-----
Find a student by first name
-----
Enter the first name: Kerolos
-----
Student first name: Kerolos
Student last name: Shenouda
Student roll number: 2
Student GPA number: 3.9
Course 1 ID: 99
Course 2 ID: 1
Course 3 ID: 55
Course 4 ID: 66
Course 5 ID: 77
-----
Student first name: Kerolos
Student last name: Gamal
Student roll number: 4
Student GPA number: 3.5
Course 1 ID: 45
Course 2 ID: 21
Course 3 ID: 55
Course 4 ID: 18
Course 5 ID: 18
```

```
Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
```

```
Enter your choice to perform the task: 4
```

```
-----
Find a student by first name
-----
```

```
Enter the first name: Ahmed
```

```
[ERROR] First name Ahmed not found
```

```
Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
```

```
Enter your choice to perform the task: 5
```

```
-----
Find students by course ID
-----
```

```
Enter the course ID: 1
```

```
-----
Student first name: Marco
Student last name: Magdy
Student roll number: 1
Student GPA number: 3.5
-----
```

```
Student first name: Kerolos
Student last name: Shenouda
Student roll number: 2
Student GPA number: 3.9
```

```
[INFO] Total number of students enrolled: 2
```



```
Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 5
-----
Find students by course ID
-----
Enter the course ID: 444
[ERROR] Course ID 444 not found

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 7
-----
Delete a student by roll number
-----
Enter the roll number: 3
[INFO] The roll number is removed successfully
```

Choose the task that you want to perform

1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit

Enter your choice to perform the task: 9

Student first name: Mazen
Student last name: Talaat
Student roll number: 5
Student GPA number: 4.0
Course 1 ID: 9
Course 2 ID: 8
Course 3 ID: 7
Course 4 ID: 6
Course 5 ID: 5

Student first name: Marco
Student last name: Magdy
Student roll number: 1
Student GPA number: 3.5
Course 1 ID: 1
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 5

Student first name: Kerolos
Student last name: Shenouda
Student roll number: 2
Student GPA number: 3.9
Course 1 ID: 99
Course 2 ID: 1
Course 3 ID: 55
Course 4 ID: 66
Course 5 ID: 77

Student first name: Kerolos
Student last name: Gamal
Student roll number: 4
Student GPA number: 3.5
Course 1 ID: 45
Course 2 ID: 21
Course 3 ID: 55
Course 4 ID: 18
Course 5 ID: 18

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 6

[INFO] The total number of students is 4
[INFO] You can add up to 50 students
[INFO] You can add 46 more students

Choose the task that you want to perform
1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit
Enter your choice to perform the task: 8

Update a student by roll number

Enter the roll number: 5

1. First name
2. Last name
3. Roll number
4. GPA
5. Courses

Enter your choice: 2

Enter the new last name: Haggag

[INFO] Updated successfully

Choose the task that you want to perform

1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit

Enter your choice to perform the task: 9

Student first name: Mazen
Student last name: Haggag
Student roll number: 5
Student GPA number: 4.0
Course 1 ID: 9
Course 2 ID: 8
Course 3 ID: 7
Course 4 ID: 6
Course 5 ID: 5

Student first name: Marco
Student last name: Magdy
Student roll number: 1
Student GPA number: 3.5
Course 1 ID: 1
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 5

Student first name: Kerolos
Student last name: Shenouda
Student roll number: 2
Student GPA number: 3.9
Course 1 ID: 99
Course 2 ID: 1
Course 3 ID: 55
Course 4 ID: 66
Course 5 ID: 77

Student first name: Kerolos
Student last name: Gamal
Student roll number: 4
Student GPA number: 3.5
Course 1 ID: 45
Course 2 ID: 21
Course 3 ID: 55
Course 4 ID: 18
Course 5 ID: 18

Choose the task that you want to perform

1. Add a student details manually
2. Add a student details from a text file
3. Find student details by roll number
4. Find student details by first name
5. Find student details by course ID
6. Find the total number of students
7. Delete student details by roll number
8. Update student details by roll number
9. Show all information
10. Exit

Enter your choice to perform the task: 10
Bye!