

Mastering Embedded System Online Diploma

[www.learn-in-depth.com](http://www.learn-in-depth.com)

## Student Management System Report First Term (Final Project 2)

Eng. Abdelaziz Maher Abdelaziz

My Profile:

[www.learn-in-depth.com/online-diploma/abdelazizmaher17499@gmail.com](http://www.learn-in-depth.com/online-diploma/abdelazizmaher17499@gmail.com)

## Problem Statement:

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

- 1) Store the First name of the student.
- 2) Store the Last name of the student.
- 3) Store the unique Roll Number for every student.
- 4) Store the GPA of every student.
- 5) Store the 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 roll number.
- 8) Update a student by roll number.
- 9) Print student data.

## Software Implementation:

### **The student Management System has:**

- Source and include files for the queue (FIFO) and Student management
- FIFO buffer is implemented as a Struct, Database is also implemented as an array of structs

Main:

- 1) Initialize FIFO queue for the database.
- 2) Interact with user through options implemented in a switch statements.

```
#include "Database.h"
#include "fifo.h"

int main()
{
    char option[40];

    if( Init_DataBaseQueue() != FIFO_NO_ERROR )
    {
        DPRINTF("\n[ERROR] DataBase Not Initialized Successfully");

        return FIFO_ERROR;
    }

    DPRINTF("\n\t <===== Welcome To The Students Management System =====>\n");
    while(1)
    {
        DPRINTF("\n=====");
        DPRINTF("\n\tChoose one of the following options");
        DPRINTF("\n1: Add The Student Details Manually");
        DPRINTF("\n2: Add The Student Details From Text File");
        DPRINTF("\n3: Find The Student Details By Roll Number");
        DPRINTF("\n4: Find The Student Details By First Name");
        DPRINTF("\n5: Find The Student Details By Course ID");
        DPRINTF("\n6: Find The Total Number Of Students");
        DPRINTF("\n7: Delete The Student Details By Roll Number");
        DPRINTF("\n8: Update The Student Details By Roll Number");
        DPRINTF("\n9: Show All Students Information");
        DPRINTF("\n10: EXIT");
        DPRINTF("\n=====");

        DPRINTF("\nEnter Option Number : ");
        gets(option);

        DPRINTF("\n=====");

        switch( atoi(option) )
        {
            case 1:
                AddStudent_Manually();
                break;
            case 2:
                AddStudent_From_File();
                break;
            case 3:
                FindBy_Roll();
                break;
            case 4:
                FindBy_FirstName();
                break;
            case 5:
                FindBy_CourseID();
                break;
            case 6:
                TotalStudents();
                break;
            case 7:
                DeleteStudent_ByRoll();
                break;
            case 8:
                UpdateStudent_Details_ByRoll();
                break;
            case 9:
                ViewStudents();
                break;
            case 10:
                exit(0);
                break;
            default:
                DPRINTF("\nWrong Option");
                break;
        }
    }

    return 0;
}
```

- FIFO and Database as structs

```

1
2 typedef struct
3 {
4     char FirstName[50];
5     char LastName[50];
6     int Roll;
7     float GPA;
8     int Course_ID[5];
9 }sStudentInfo;
10
11
12 #define element_type    sStudentInfo
13
14
15 typedef struct
16 {
17     element_type *base;
18     element_type *head;
19     element_type *tail;
20     uint8_t count;
21     uint32_t length;
22 }FIFO_t;
23
24 typedef enum
25 {
26     FIFO_NO_ERROR,
27     FIFO_ERROR,
28     FIFO_NULL,
29     FIFO_EMPTY,
30     FIFO_FULL
31 }FIFO_state;
32
33
34 FIFO_state FIFO_Init(FIFO_t* new ,element_type* pBuffer, uint32_t Length);
35 FIFO_state FIFO_Enqueue(FIFO_t* add , element_type item );
36 FIFO_state FIFO_Dequeue(FIFO_t* remove, element_type item);
37 FIFO_state FIFO_IsFull(FIFO_t* check);
38 void FIFO_print(FIFO_t* print);
39
40 #endif /* FIFO_H_ */

```

- APIs for the management of the database

```
#ifndef DATABASE_H_
#define DATABASE_H_

#define UNVALID_POS      -1
#define UNVALID_ID       -2
|
#define RESULT_NOT_FOUND  0
#define RESULT_FOUND      1

#define STR_IDENTICAL     0

#define DATA_EXIST       1

int Init_DataBaseQueue();
void AddStudent_From_File();
void AddStudent_Manually();
void DeleteStudent_ByRoll();
void ViewStudents();
void TotalStudents();
void FindBy_Roll();
void FindBy_CourseID();
void FindBy_FirstName();
void UpdateStudent_Details_ByRoll();

#endif /* DATABASE_H_ */
```

- Fill student info that entered manually.

```
int fill_the_record(sStudentInfo* data)
{
    char temp_text[50];
    int i;

    DPRINTF("\nStudent Roll Number: ");
    gets(temp_text);

    if( CheckValid_ID( atoi(temp_text) ) == UNVALID_ID )
    {
        DPRINTF("\n[ERROR] Roll Number %d Is Already Taken",atoi(temp_text) );

        return UNVALID_ID; // None valid id
    }

    data->Roll = atoi(temp_text);

    DPRINTF("\nStudent First Name: ");
    gets(data->FirstName);

    DPRINTF("\nStudent Last Name: ");
    gets(data->LastName);

    DPRINTF("\nStudent GPA: ");
    gets(temp_text);
    data->GPA = atof(temp_text);

    DPRINTF("\nEnter Course ID of each Course: ");
    for(i = 0 ; i < 5 ; i++ )
    {
        DPRINTF("\nCourse %d ID : ", i+1 );

        gets(temp_text);
        data->Course_ID[i] = atoi(temp_text);
    }

    return 0;
}
```

- Check if student id is valid or not.

```
int CheckValid_ID(int id)
{
    int i;

    for( i = 0 ; i < DataBase.count ; i++ )
    {
        if( (DataBase_BUFF[i].Roll ) == id )
            return UNVALID_ID; //None valid id
    }

    return 0;
}
```



- Add students to the database from a text file.

```
void AddStudent_From_File()
{
    int i;

    FILE *Student_File = fopen("Student_List.txt", "r" );

    if(Student_File == NULL){
        printf("\t[ERROR]File opening Failed\n");
        return;
    }

    while( !feof(Student_File) && FIFO_IsFull(&DataBase) != FIFO_FULL )
    {
        sStudentInfo pAddedStudent ;

        fscanf(Student_File,"%d", &(pAddedStudent.Roll) );
        fscanf(Student_File,"%s", (pAddedStudent.FirstName) );
        fscanf(Student_File,"%s", (pAddedStudent.LastName) );
        fscanf(Student_File,"%f", &(pAddedStudent.GPA) );

        for(i = 0 ; i < 5 ; i++ )
        {
            fscanf(Student_File,"%d", &(pAddedStudent.Course_ID[i]) );
        }

        //Check its validity
        if( CheckValid_ID( pAddedStudent.Roll ) == UNVALID_ID )
        {
            DPRINTF("\n[ERROR] Roll Number %d Is Already Taken", pAddedStudent.Roll );

            continue;
        }

        FIFO_Enqueue(&DataBase , pAddedStudent );
        DPRINTF("\n[INFO] Student %d Is Added Successfully", pAddedStudent.Roll);
    }
    fclose(Student_File);

    DPRINTF("\n[INFO] Student Details Are Added Successfully" );
}
```



Student\_List - Notepad

File Edit Format View Help

```
1 Abdelaziz Maher 3.6 1 2 3 4 5
2 Mohamed Emad 2.0 4 10 12 13 15
3 Mohamed Hassan 2.1 1 2 6 7 8
4 Hussein Sherief 3.3 9 10 11 12 13
5 Ziad Maher 3.9 4 5 6 7 12
```



- Add student by taking info from user.

```
void AddStudent_Manually()
{
    if( FIFO_IsFull(&DataBase) != FIFO_FULL )
    {
        if( fill_the_record(&DataBase_BUFF[DataBase.count]) == DATA_EXIST )
        {
            DPRINTF("\n[ERROR] Roll Number %d Is Already Taken", DataBase_BUFF[DataBase.count].Roll );
        }
        else
        {
            FIFO_Enqueue(&DataBase , DataBase_BUFF[DataBase.count] );
            DPRINTF("\n[INFO] Student Details Are Added Successfully" );
        }
    }
    else
    {
        DPRINTF("\n[ERROR] DataBase Queue Is FULL:");
    }
}
```

- Delete student by getting his ID.

```
void DeleteStudent_ByRoll()
{
    int ID_ToDelete;
    int i , Pos = UNVALID_POS;
    DPRINTF("\nEnter The Roll Number You Want To Delete:");
    scanf("%d",&ID_ToDelete);

    for(i = 0 ; i < DataBase.count ; i++)
    {
        if(DataBase_BUFF[i].Roll == ID_ToDelete)
        {
            Pos = i;
            break;
        }
    }

    if( Pos == UNVALID_POS)
    {
        DPRINTF("\n[ERROR] This Roll Number %d Is Not Found",ID_ToDelete);
        return;
    }

    for(i = Pos ; i < DataBase.count ; i++ )
        DataBase_BUFF[i] = DataBase_BUFF[i+1];

    DataBase.head--;
    DataBase.count--;

    DPRINTF("\n[INFO] This Roll Number Is Removed Successfully");
}
```

- Show students info.

```
void ViewStudents()
{
    int i, j;

    DPRINTF("\n\t=====Students Info In The DataBase=====\\n");
    for(i = 0; i < DataBase.count ; i++ )
    {
        DPRINTF("\\n=====Student %d Info===== \\n", i+1);

        DPRINTF("Student First Name: %s \\n", DataBase_BUFF[i].FirstName);
        DPRINTF("Student Last Name: %s \\n", DataBase_BUFF[i].LastName);
        DPRINTF("Student Roll Number: %d \\n", DataBase_BUFF[i].Roll);
        DPRINTF("Student GPA: %f \\n", DataBase_BUFF[i].GPA);
        for(j = 0 ; j < 5 ; j++ )
        {
            DPRINTF("Course %d ID : %d \\n", j+1 , DataBase_BUFF[i].Course_ID[j]);
        }
    }
}
```

- Print the total number of students.

```
void TotalStudents()
{
    DPRINTF("\\n[INFO] The Total Number Of students is %d",DataBase.count);
    DPRINTF("\\n[INFO] You Can Add Up To 50 Students\\n");
    DPRINTF("\\n[INFO] You Can Add %d More Students", 50-DataBase.count );
}
```

- Find a student by searching for his ID.

```
void FindBy_Roll()
{
    int Roll_ToFind, result = RESULT_NOT_FOUND;
    int i , j;

    DPRINTF("\nEnter The Roll Number Of The Student:");
    scanf("%d",&Roll_ToFind);

    for(i = 0 ; i < DataBase.count ; i++)
    {
        result = DataBase_BUFF[i].Roll == Roll_ToFind;
        if( result == RESULT_FOUND )
        {
            DPRINTF("\n\t=====The Student Details=====\\n");

            DPRINTF("Student First Name: %s \\n", DataBase_BUFF[i].FirstName);
            DPRINTF("Student Last Name: %s \\n", DataBase_BUFF[i].LastName);
            DPRINTF("Student Roll Number: %d \\n", DataBase_BUFF[i].Roll);
            DPRINTF("Student GPA: %f \\n", DataBase_BUFF[i].GPA);

            for(j = 0 ; j < 5 ; j++ )
            {
                DPRINTF("\\nCourse %d ID : %d ", j+1 , DataBase_BUFF[i].Course_ID[j]);
            }

            break;
        }
    }

    if( result == RESULT_NOT_FOUND)
    {
        DPRINTF("\\n[ERROR] This Roll Number %d Is Not Found",Roll_ToFind);
        return;
    }
}
```

- Find a student by searching for his Course ID.

```
void FindBy_CourseID()
{
    int CourseID_ToFind, flag = 0;
    int i , j;

    DPRINTF("\nEnter The Course ID Of The Student:");
    scanf("%d",&CourseID_ToFind);

    for(i = 0 ; i < DataBase.count ; i++)
    {
        for(j = 0 ; j < 5 ; j++)
        {
            if( DataBase_BUFF[i].Course_ID[j] == CourseID_ToFind )
            {
                DPRINTF("\n\t=====The Student Details=====\\n");

                DPRINTF("Student First Name: %s \\n", DataBase_BUFF[i].FirstName);
                DPRINTF("Student Last Name: %s \\n", DataBase_BUFF[i].LastName);
                DPRINTF("Student Roll Number: %d \\n", DataBase_BUFF[i].Roll);
                DPRINTF("Student GPA: %f \\n", DataBase_BUFF[i].GPA);

                for(j = 0 ; j < 5 ; j++ )
                {
                    DPRINTF("\\nCourse %d ID : %d ", j+1 , DataBase_BUFF[i].Course_ID[j]);
                }
                flag = 1;
                break;
            }
        }
    }

    if( flag == 0)
    {
        DPRINTF("\\n[ERROR] This Course ID %d Is Not Found",CourseID_ToFind);
        return;
    }
}
```

- Find a student by searching for his First name.

```
void FindBy_FirstName()
{
    char Name_ToFind[50];
    int i , j, result = RESULT_NOT_FOUND;

    DPRINTF("\nEnter The First Name Of The Student:");
    scanf("%s",Name_ToFind);

    for(i = 0 ; i < DataBase.count ; i++)
    {
        result = strcmp(DataBase_BUFF[i].FirstName, Name_ToFind);
        if( result == STR_IDENTICAL )
        {
            DPRINTF("\n\t=====The Student Details=====\\n");

            DPRINTF("Student First Name: %s \\n", DataBase_BUFF[i].FirstName);
            DPRINTF("Student Last Name: %s \\n", DataBase_BUFF[i].LastName);
            DPRINTF("Student Roll Number: %d \\n", DataBase_BUFF[i].Roll);
            DPRINTF("Student GPA: %f \\n", DataBase_BUFF[i].GPA);

            for(j = 0 ; j < 5 ; j++ )
            {
                DPRINTF("\\nCourse %d ID : %d ", j+1 , DataBase_BUFF[i].Course_ID[j]);
            }

            break;
        }
    }

    if( result != STR_IDENTICAL)
    {
        DPRINTF("\\n[ERROR] This First Name %s Is Not Found",Name_ToFind);
        return;
    }
}
```



- Update student info by getting his ID.

```

void UpdateStudent_Details_ByRoll()
{
    int Roll_ToUpdate, result = RESULT_NOT_FOUND;
    int Updated_Info;
    int i , j;
    char temp_text[10];

    DPRINTF("\nEnter The Roll Number To Update The Entry:");
    scanf("%d",&Roll_ToUpdate);

    for(i = 0 ; i < DataBase.count ; i++)
    {
        result = DataBase_BUFF[i].Roll == Roll_ToUpdate;
        if( result == RESULT_FOUND )
        {
            DPRINTF("\n\t=====The Student Details=====\\n");

            DPRINTF("Student First Name: %s \\n", DataBase_BUFF[i].FirstName);
            DPRINTF("Student Last Name: %s \\n", DataBase_BUFF[i].LastName);
            DPRINTF("Student Roll Number: %d \\n", DataBase_BUFF[i].Roll);
            DPRINTF("Student GPA: %f \\n", DataBase_BUFF[i].GPA);

            for(j = 0 ; j < 5 ; j++ )
            {
                DPRINTF("\\nCourse %d ID : %d ", i+1 , DataBase_BUFF[i].Course_ID[j]);
            }

            break;
        }
    }

    if( result == RESULT_NOT_FOUND )
    {
        DPRINTF("\\n[ERROR] This Roll Number %d Is Not Found",Roll_ToUpdate);
        return;
    }

    DPRINTF("\\n=====Choose INFO To Update=====");
    DPRINTF("\\n(1)First Name");
    DPRINTF("\\n(2>Last Name");
    DPRINTF("\\n(3)GPA");
    DPRINTF("\\n(4)Registered Courses");
    DPRINTF("\\nOption: ");
    scanf("%d",&Updated_Info);

    switch(Updated_Info)
    {
        case 1:
            DPRINTF("\\nEnter The New First Name: ");
            scanf("%s",DataBase_BUFF[i].FirstName);

            DPRINTF("\\n[INFO] UPDATED SUCCESSFULLY");
            break;
        case 2:
            DPRINTF("\\nEnter The New Last Name:");
            scanf("%s",DataBase_BUFF[i].LastName);

            DPRINTF("\\n[INFO] UPDATED SUCCESSFULLY");
            break;
        case 3:
            DPRINTF("\\nEnter The New GPA:");
            scanf("%f",&DataBase_BUFF[i].GPA);

            DPRINTF("\\n[INFO] UPDATED SUCCESSFULLY");
            break;
        case 4:
            DPRINTF("\\nEnter The New Courses IDs:");
            for(j = 0 ; j < 5 ; j++ )
            {
                DPRINTF("\\nCourse %d ID : ", j+1 );

                gets(temp_text);
                DataBase_BUFF[i].Course_ID[j] = atoi(temp_text);
            }
            DPRINTF("\\n[INFO] UPDATED SUCCESSFULLY");
            break;
        default:
            DPRINTF("\\n[ERROR] Enter Right Option Please");
    }
}

```

# Simulation:

```
<<===== Welcome To The Students Management System =====>>

=====
      Choose one of the following options
1: Add The Student Details Manually
2: Add The Student Details From Text File
3: Find The Student Details By Roll Number
4: Find The Student Details By First Name
5: Find The Student Details By Course ID
6: Find The Total Number Of Students
7: Delete The Student Details By Roll Number
8: Update The Student Details By Roll Number
9: Show All Students Information
10: EXIT
=====
Enter Option Number : 1

=====
Student Roll Number: 1

Student First Name: Alaa

Student Last Name: Nabeel

Student GPA: 3.56

Enter Course ID of each Course:
Course 1 ID : 6

Course 2 ID : 7

Course 3 ID : 9

Course 4 ID : 10

Course 5 ID : 12

[INFO] Student Details Are Added Successfully
=====
      Choose one of the following options
```



```
=====
      Choose one of the following options
1: Add The Student Details Manually
2: Add The Student Details From Text File
3: Find The Student Details By Roll Number
4: Find The Student Details By First Name
5: Find The Student Details By Course ID
6: Find The Total Number Of Students
7: Delete The Student Details By Roll Number
8: Update The Student Details By Roll Number
9: Show All Students Information
10: EXIT
=====
```

```
Enter Option Number : 2
```

```
[
```

```
=====
[ERROR] Roll Number 1 Is Already Taken
[INFO] Student 2 Is Added Successfully
[INFO] Student 3 Is Added Successfully
[INFO] Student 4 Is Added Successfully
[INFO] Student 5 Is Added Successfully
[INFO] Student Details Are Added Successfully
=====
```

=====

Choose one of the following options

- 1: Add The Student Details Manually
- 2: Add The Student Details From Text File
- 3: Find The Student Details By Roll Number
- 4: Find The Student Details By First Name
- 5: Find The Student Details By Course ID
- 6: Find The Total Number Of Students
- 7: Delete The Student Details By Roll Number
- 8: Update The Student Details By Roll Number
- 9: Show All Students Information
- 10: EXIT

=====

Enter Option Number : 3

=====

Enter The Roll Number Of The Student:1

[

=====The Student Details=====

Student First Name: Alaa  
Student Last Name: Nabeel  
Student Roll Number: 1  
Student GPA: 3.560000

Course 1 ID : 6  
Course 2 ID : 7  
Course 3 ID : 9  
Course 4 ID : 10  
Course 5 ID : 12

=====

Enter Option Number : 4

=====

Enter The First Name Of The Student:Abdelaziz

[ERROR] This First Name Abdelaziz Is Not Found

=====

Choose one of the following options

- 1: Add The Student Details Manually
- 2: Add The Student Details From Text File
- 3: Find The Student Details By Roll Number
- 4: Find The Student Details By First Name
- 5: Find The Student Details By Course ID
- 6: Find The Total Number Of Students
- 7: Delete The Student Details By Roll Number
- 8: Update The Student Details By Roll Number
- 9: Show All Students Information
- 10: EXIT

=====

Enter Option Number : 4

=====

Enter The First Name Of The Student:Mohamed

[

=====The Student Details=====

Student First Name: Mohamed

Student Last Name: Emad

Student Roll Number: 2

Student GPA: 2.000000

Course 1 ID : 4

Course 2 ID : 10

Course 3 ID : 12

Course 4 ID : 13

Course 5 ID : 15

=====

-----  
Enter Option Number : 5

=====  
Enter The Course ID Of The Student:10

[  
          =====The Student Details=====

Student First Name: Alaa  
Student Last Name: Nabeel  
Student Roll Number: 1  
Student GPA: 3.560000

Course 1 ID : 6  
Course 2 ID : 7  
Course 3 ID : 9  
Course 4 ID : 10  
Course 5 ID : 12

          =====The Student Details=====

Student First Name: Mohamed  
Student Last Name: Emad  
Student Roll Number: 2  
Student GPA: 2.000000

Course 1 ID : 4  
Course 2 ID : 10  
Course 3 ID : 12  
Course 4 ID : 13  
Course 5 ID : 15

          =====The Student Details=====

Student First Name: Hussein  
Student Last Name: Sherief  
Student Roll Number: 4  
Student GPA: 3.300000

Course 1 ID : 9  
Course 2 ID : 10  
Course 3 ID : 11  
Course 4 ID : 12  
Course 5 ID : 13

=====



=====

Choose one of the following options

- 1: Add The Student Details Manually
- 2: Add The Student Details From Text File
- 3: Find The Student Details By Roll Number
- 4: Find The Student Details By First Name
- 5: Find The Student Details By Course ID
- 6: Find The Total Number Of Students
- 7: Delete The Student Details By Roll Number
- 8: Update The Student Details By Roll Number
- 9: Show All Students Information
- 10: EXIT

=====

Enter Option Number : 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 one of the following options

- 1: Add The Student Details Manually
- 2: Add The Student Details From Text File
- 3: Find The Student Details By Roll Number
- 4: Find The Student Details By First Name
- 5: Find The Student Details By Course ID
- 6: Find The Total Number Of Students
- 7: Delete The Student Details By Roll Number
- 8: Update The Student Details By Roll Number
- 9: Show All Students Information
- 10: EXIT

=====

Enter Option Number : 7

=====

Enter The Roll Number You Want To Delete:1

[

[INFO] This Roll Number Is Removed Successfully

=====

Choose one of the following options

- 1: Add The Student Details Manually
- 2: Add The Student Details From Text File
- 3: Find The Student Details By Roll Number
- 4: Find The Student Details By First Name
- 5: Find The Student Details By Course ID
- 6: Find The Total Number Of Students
- 7: Delete The Student Details By Roll Number
- 8: Update The Student Details By Roll Number
- 9: Show All Students Information
- 10: EXIT

=====

Enter Option Number : 8

=====

Enter The Roll Number To Update The Entry:2

=====The Student Details=====

Student First Name: Mohamed

Student Last Name: Emad

Student Roll Number: 2

Student GPA: 2.000000

Course 1 ID : 4

Course 1 ID : 10

Course 1 ID : 12

Course 1 ID : 13

Course 1 ID : 15

=====Choose INFO To Update=====

(1)First Name

(2)Last Name

(3)GPA

(4)Registered Courses

Option: 2

Enter The New Last Name:Mazen

[  
[INFO] UPDATED SUCCESSFULLY

-----

Enter Option Number : 9

[

=====

=====Students Info In

=====Student 1 Info=====

Student First Name: Mohamed

Student Last Name: Mazen

Student Roll Number: 2

Student GPA: 2.000000

Course 1 ID : 4

Course 2 ID : 10

Course 3 ID : 12

Course 4 ID : 13

Course 5 ID : 15

=====Student 2 Info=====

Student First Name: Mohamed

Student Last Name: Hassan

Student Roll Number: 3

Student GPA: 2.100000

Course 1 ID : 1

Course 2 ID : 2

Course 3 ID : 6

Course 4 ID : 7

Course 5 ID : 8

=====Student 3 Info=====

Student First Name: Hussein

Student Last Name: Sherief

Student Roll Number: 4

Student GPA: 3.300000

Course 1 ID : 9

Course 2 ID : 10

Course 3 ID : 11

Course 4 ID : 12

Course 5 ID : 13

=====Student 4 Info=====

Student First Name: Ziad

Student Last Name: Maher

Student Roll Number: 5

Student GPA: 3.900000

Course 1 ID : 4

Course 2 ID : 5

Course 3 ID : 6

Course 4 ID : 7

Course 5 ID : 12

=====



-----  
Choose one of the following options

- 1: Add The Student Details Manually
- 2: Add The Student Details From Text File
- 3: Find The Student Details By Roll Number
- 4: Find The Student Details By First Name
- 5: Find The Student Details By Course ID
- 6: Find The Total Number Of Students
- 7: Delete The Student Details By Roll Number
- 8: Update The Student Details By Roll Number
- 9: Show All Students Information
- 10: EXIT

=====

Enter Option Number : 10

[

=====