***SRS DOCUMENTATION***

***Lab Batch:-45***

***Topic:- “STUDENT DATABASE MANAGEMENT”***

***…(plz open file in MS word to view the flowchart) …………………………………………………………...........***

***CONTENTS***

***1.Introduction***

***2.Structures used in the project***

***3.Files Used in the project***

***4.Assumptions made in the program***

***5.Innovations done in the project***

***6.Global and local variables Used***

***7.Flow Chart of the project***

***8.Header files Included***

***9.Detailed Description Of Functions called and where are they called***

***10.What every individual in batch is handling and contributions of them in this Documentation.***

***11.Limitations of the project***

***12.References/Bibliography***

1. ***Introduction***

***Handling one of the major practical, applicative and highly useful topic , the project topic of our batch reads:- “STUDENT DATABASE MANAGEMENT”.....***

***As is all suggested by its name , it presents how to manage a large scale database for students. It typically follows IITB Freshie year format where every student has a division , tutorial batch and there are no department specific courses for any of them.***

***Typically a database software implementing and handling student data, is used both by admnistrator and student . Administrator has the right to add new student , delete a student's record , enter new marks and modify them... But such rights are not available for a student user, for whom only displaying of few options are available where he may wish to***

***see his marksheet***

***see what courses he/she is enrolled in and credits meant for those.***

***Thus our project implements both the aspects whether the user is an administrator or a student. Hence Display of menu would be driven by the password entry one does that designates him with administrator or a student...***

***The typical storage of data for students is done using 4 files, all implementing the DBMS concept of primary key .***

***Another concept used in our project is the usage of Ezwindows for entire display operation performed. Since Ezwindows donot implement entry of user data on SimpleWindow screen thus we assume switch between Terminal screen and SimpleWindow***

***screen every time a input has to be given by the user.***

***The project is meant to handle large scale data for students but is meant to provide interaction with user. Thus we firstly direct few of records from other file and store them on files to be used by us. Now everytime a new student would be entered that entry would direct input from Keyboard not from file.***

***Rest all goes well if its well implemented. Project structure as is decided comes in subsequent pages.***

***Hope u Enjoy…..***

***……………………………………………………………………………………***

***STRUCTURES TO BE USED***

1)

struct student\_info

{

char Name[25];

char Roll\_no[12];

char Branch[22];

short int Division;

struct DOB

{

short int date;

short int month;

short int year;

}dob;

short int Hostel\_no;

short int Room\_no;

char Home\_state[22];

char extra\_act[5]; // to hold what extra course he is enrolled in like NSS/NSO

char blood\_group[5];

char password[22];

}student;

2) struct result\_info

{

char Roll\_no[12];

int Marks[6];

char extra\_act[];

float SPI;

}result;

3) struct course\_info

{

short int Division;

struct course\_details

{

char course\_name[10];

char tutor\_name[22];

short int credits;

}subject[6];

}course;

4) struct Index\_info

{

char Roll\_no[12];

short int Record\_no;

}index;

…………………………………………………………………………………

***Files to be used***

1)Student file

This implements records of student\_info structure.

This is using Binary implementation.

2)Result file

This implements records of result\_info structure.

This is using Binary implementation.

3)Course file

This implements records of course\_info structure.

This is using Binary implementation.

4)Index file

This implements records of index\_info structure.

This is using Binary implementation.

This file is meant for implementing indexing in our program by having students roll number and his record number in student file as its one record…

…………………………………………………………………………………………

***Assumptions in our project***

* Data is meant for first year students only….

Thus Division concept has been used. Also extra course for NSS/NSO etc even are used.

* In D3 division we are not implementing that electrical students will have a different course EE111 than HS101 of civil students. This is because that would harm the symmetry of Course file and would be a bit difficult to handle.
* We have to switch between Terminal and SimpleWindow everytime input is needed.
* We have used Terminal and simple window both for output display not just SimpleWindow…..

…………………………………………………………………………………

***Innovations in our project***

* We have implemented a “QUERIES” option in our project that handles with the most common Queries that everyone has regarding Student Database like Highest marks Display and others as u will see.
* (Idea of Karunesh Jigyasu of labbatch 15) We have implemented self review file concept where everyone needs to submit what all he/she has done for the project and on basis of that what marks should be given to him/her.

…………………………………………………………………………

***Global and local variables used***

1. Global variables:-

* 4 variables of the global structures defined will be there each of other type. Those variables will be used everywhere we need to read that file or write onto that.
* char admin\_password[]=”81admin\_iitb90”;

Meant for defining password for administrator that will be common for all.

* char pass\_entry[20] // variable to take input of password anywhere and everywhere in the program.
* Char roll\_entry[11]
* Int a // to take input of which numbered menu to open

1. Local variables would be described as the function is described.

…………………………………………………………………………………….

***Flowchart of our project***

***Header files included***

* Ezwin.h
* Cassert
* Cstring
* Cstdio
* Iostream
* Cmath
* SimpleWindow.h

……………………………………………………………………………………………………….

***Description of Functions used***

1. **ApiMain()**

**Variables used:-** SimpleWindow Main\_Menu

//object that represent window screen

**Function outline:-**

* Using RenderText function of Main\_Menu I’ll display main menu as such with a number in front of each menu.
* User inputs the value of number in front of menu he want to choose on the terminal window.
* According to the choices using Switch statement I will call corresponding function.

Case choice1 administrator\_login() will be called

Case choice2 student\_login() will be called

Case choice3 program is exited

2)***Result\_processing\_menu()***

FUNCTION NAME :- result\_processing\_menu()

WORK OF THE FUNCTION :- It will display the menu to the user when he chooses the

result\_processing\_menu and works further as per the choice entered by the user.

VARIABLES USED :- choice (to take the choice from the user),win(to open a window)

ALGORITHM:-

First the function will open a new window through a SimpleWindow object

win.

The following choices will be put on the window through the function

rendertext().

1 . Enter new marks

2 . Modify existing marks

3 . Exit

The user will be asked to enter a choice.

The choice entered by the user will be stored in the choice variable.

It’ s validity will be checked and if found invalid the user will be asked to

enter it again.

The choice variable will be operated upon a switch statement that will invoke

appropriate functions as per the choice entered by the user.

3)***QUERIES***

FUNCTION NAME :- queries()

WORK OF THE FUNCTION :- It will display the menu to the user when he chooses the

queries option and works further as per the choice entered by the user.

VARIABLES USED :- choice (to take the choice from the user),win(to open a window)

ALGORITHM:-

First the function will open a new window through a SimpleWindow object

win.

The following choices will be put on the window through the function

rendertext().

1 . Query1

2 . Query2

3 . Query3

4 . Query4

5 . Query5

6 . Query6

7 . Exit

The user will be asked to enter a choice.

The choice entered by the user will be stored in the choice variable.

It’ s validity will be checked and if found invalid the user will be asked to

enter it again.

The choice variable will be operated upon a switch statement that will invoke

appropriate functions as per the choice entered by the user.

FUNCTION NAME :- display\_menu()

WORK OF THE FUNCTION :- It will display the menu to the user when he chooses the

display menu and works further as per the choice entered by the user.

VARIABLES USED :- choice (to take the choice from the user),win(to open a window)

ALGORITHM:-

•

•

•

•

•

•

First the function will open a new window through a SimpleWindow object

win.

The following choices will be put on the window through the function

rendertext().

1 . Student file display

2 . Course file display

3 . Marksheet of student

4 . Queries

5 . Exit

The user will be asked to enter a choice.

The choice entered by the user will be stored in the choice variable.

It’ s validity will be checked and if found invalid the user will be asked to

enter it again.

The choice variable will be operated upon a switch statement that will invoke

appropriate functions as per the choice entered by the user.

**( query1) FUNCTION FOR BLOOD GROUP WISE STUDENT COUNT**

One of the programs to be written is bloodgroup wise student count.It gives the number of students under a particular blood group whensoever asked by the user.

INPUT

The user chooses the subheader 'blood group wise student count' in the menu under queries.

OUTPUT

The output is the number of students belonging to a particular blood group.

STRUCTURES OF DIFFERENT FILES

STUDENT STRUCTURE

{

char name[20];

char roll\_no[11];

char branch[15];

short int divison // 1 for d1

2 for d2

DOB date\_of\_birth;

short int hstl\_no,room\_no,

char homestate[15];

char additional\_course[4]; //ncc,nso,nss

char blood\_group[5];

char password[15];

}

struct DOB

{ int date;month,year;}

COURSE STRUCTURE

{

short int divison;

substructure a[6];

}

struct substructure

{

char course\_name[6];

char instructor\_name[20];

short int course\_credits;

}

OUTLINE OF THE PROGRAM

1. I wish to make two arrays. one of them is a two dimensional string array and the other is an integer array.

2. The dimensions of the string array is 8\*5. I feed or initialize the array row wise by the eight different blood groups.

3. The other array is an integer array containing 8 elements corresponding to the number of students.The values are initialized to zero in this array

4. I want to estabilish a one to one correspondence between the array of strings and array of integers.The first row of the multidimensional string which contains a blood group corresponds to the number of students with that blood group .

5. The file of student data contains a large number of records.Each record contains data of the particular student in the form of a structure.So the idea is to copy the fixed length record in to a structure and access the

bloodgroup in that.

6.Now i create a file pointer named student\_data. i open the binary file of student data pointing the pointer to it with 'fopen' statement.I set the internal pos variable to the initial position with the help of 'fseek' statement.

7. Now i set up a while loop and i set the internal pos varible in motion as follows. i say 'while(!feof) and copy the first record in to a structure which i declare in the beginning of the program. using the 'fread' statement.

I use the sizeof function to know the length of the fixed length record.initially since the pointer is at the initial position it will start reading. It copies the data of length equal to the size of student\_info structure in

to the structure which i declared already.

fread(struct student\_info var,record\_size,1,student\_info.dat)

8. Now in the loop i set up another loop to check if the blood group i.e var.bloodgroup matches with any of the blood groups of the multidimensional string array.And if it is found that they have matched i increment the corresponding element in the integer array by 1.

9.Now one of the elemants of the array is incremented by one. Since the pointer pos is now at the beginning of the second record it copies the second record in to the structure var now because of the while loop.

10. So after each record is read the array element of the corresponding bloodgroup is incremented.So at each and every stage the number of students under a particular blood group is being recorded.

11. Once the pointer reaches the end of the file the loop stops and by then the array will be containing the number of students having a particular blood group.

12.Next the mandatory things like closing the file ,returning values,etc...are to be taken care of.

(query5) DOCUMENTATION OF THE SECOND PROGRAM

The second program is regarding divison wise course and instructor.

when the student asks for the courses he or she has been regisered this program provides data about the courses and the instructors in charge.

INPUT

Divison to which the student belongs.

OUTPUT

The courses for which he or she has been registered and the instructors in charge.

STRUCTURES OF DIFFERENT FILES

STUDENT STRUCTURE

{

char name[20];

char roll\_no[11];

char branch[15];

short int divison // 1 for d1

2 for d2

DOB date\_of\_birth;

short int hstl\_no,room\_no,

char homestate[15];

char additional\_course[4]; //ncc,nso,nss

char blood\_group[5];

char password[15];

}

struct DOB

{ int date;month,year;}

COURSE STRUCTURE

{

short int divison;

substructure a[6];

}

struct substructure

{

char course\_name[6];

char instructor\_name[20];

short int course\_credits;

}

OUTLINE OF THE PROGRAM

1.Now i create a file pointer named course. i open the binary file , course\_file.dat pointing the pointer to it with 'fopen' statement.I set the internal pos variable to the initial position with the help of 'fseek' statement.

2.I ask the user to give the divison whose courses are required by the user.I take the divison as the input.

3.Now i set up a while loop and i set the internal pos varible in motion as follows. i say 'while(!feof) and copy the first record in to a structure which i declare in the beginning of the program. using the 'fread' statement.

I use the sizeof function to know the length of the fixed length record.initially since the pointer is at the initial position it will start reading. It copies the data of length equal to the size of student\_info structure in to the structure which i declared already.

fread(struct course\_file var,record\_size,1,course\_file.dat)

4.Now i check if the student is from d1 or d2 or d3 or d4.Once the divison is known the courses are read by the pointer.

5.The course file has a substructure component as can be seen above. As the structure of the file is simply designed i read the elements of each substructure and output it to the user.

**(query 2) PROGRAM FOR HOSTELWISE STATE STRENGTH**

There will be a structure in the program which will contain all the major informations of each hostel The details will be:

(1) name

(2) roll number

(3) branch

(4) division

(5) date\_of\_birth

(6) hostel\_number

(7) home\_state

(8) blood\_group

(9) password

There is a binary file containing all these details of the student. There is a structure in program containing all the details of the student.

The name of the structure is studentinfo. The use of the array hostel\_state[20] is made(studentinfo hostel\_state[20]). hostel\_state[i].name contains students' name,in the same way hostel\_state[i].hostel\_number contains students' hostel number and hostel\_state[i].home\_state contains

students' home state information.

A pointer(\*fp) is used in the file. while(!feof(fp)), sequentially each line of the file is read using fread(&studentinfo1, sizeof(studentinfo), 1,fp). Intially the integers 'row' , 'coloumn' , 'a' , i ,j are 0. An int array statedata[6][5] is made.

Initially all array elements are 0. We use:

for(i=0, i<6, i++) {

for(j=0, j<5, j++){ (statedata[i][j]=0;) } }

Compare the state string as shown:

if(hostel\_state.home\_state=="maharashtra")

{ row=0;}

if(hostel\_state.home\_state=="UP")

{ row=1;}

if(hostel\_state.home\_state=="MP")

{ row=2;}

if(hostel\_state.home\_state=="AP")

{ row=3;}

if(hostel\_state.home\_state=="rajasthan")

{ row=4;}

else

{ row=5;}

The hostel number of the student is equated to 'a' (a=hostel\_state.hostel\_number;). Now the switch condition is used to

determine coloumn from 'a'.

For the cases a=1,2,3,4 coloumn=a-1 and for case a=10 coloumn=4.

Everytime on reading a line, increment will be done in that statedata[row][coloumn] by 1.

To get the desired output we need to have the cout statements in the following manner:

cout<<"hostel number 1 2 3 4 10"<<endl;

cout<<"maharashtra ";

for(i=0,i<5,i++){

cout<<" "<<statedata[0][i]<<" "<<endl;

}

cout<<"UP ";

for(i=0,i<5,i++){

cout<<" "<<statedata[1][i]<<" "<<endl;

}

cout<<"MP ";

for(i=0,i<5,i++){

cout<<" "<<statedata[2][i]<<" "<<endl;

}

cout<<"AP ";

for(i=0,i<5,i++){

cout<<" "<<statedata[3][i]<<" "<<endl;

}

cout<<"rajasthan ";

for(i=0,i<5,i++){

cout<<" "<<statedata[4][i]<<" "<<endl;

}

cout<<"others ";

for(i=0,i<5,i++){

cout<<" "<<statedata[5][i]<<" "<<endl;

}

By this the user will get the state wise strengths of hostels 1, 2, 3, 4, 10.

(**query3) division wise highest marks:**

in this part of coding the user will enquire about the topper's name and marks in a particular division and course of his/her own choice.

so first lets define the input user variables as int userinputfordiv[2]//division as 1,2,3,4//,char

userinputforcourse[6]//course name as ma105,ch103,cs101 etc.//

now as the course structure and substructure have been designed, we have

struct course

{ short int Division;

substructure a[6];

};var;

struct substructure

{ char course\_name[6];

char instructor\_name[20];

short int course\_credits;

};

now i would open the course file,

FILE=\*fp

function:fp=fopen("course.dat","r");//opens the course file in read mode//

now define a variable var and t,

function:fread(&var,sizeof(var),1.0,fp);//reads the course file and directs its

file pointer "fp" to the address of

var//

now using the if-else ladder we can obtain the index of the chosen course in result file:

function: if(var.division==userinputfordiv[2])//matching the division in course

file with user input//

for(i=0,i<6,i++)//as there are 6 courses in each division//

{if(strcmp(var.a[i].course\_name,userinputforcourse[6])//it compares

both the strings, ie course name stored in course

file with the course input from the user//

t=i;//stores the index of the userinputcourse in a variable t//

else cout<<"the above course is not taught in this

division"<<endl;//as it may be possible that the user is

unaware that some particular course is not taught in a particular

division for eg. user may ask for highest marks in ph105 in d3 which not acceptable//

}

now close the course file using the function fclose,

now i would open the student file using fopen,

define an array[j] corresponding to the student structure,

student array[100];

]

if (array[j].division==userinputfordiv[6])//it compares the division of a student

with the user input//

j++;

now i will open the result file and find the corresponding marks[t]

now define a variable maxmarks and put the value of marks[t] in the variable

maxmarks

maxmarks=marks[t];

now initially the first record no. at which the student division matches with theuser input, that student's marks would be stored in maxmarks ,

but then as we go on checking ahead we will compare the new value of marks[t] withthe existing maxmarks,

if maxmarks<marks[t] then replace the value of maxmarks with the new marks[t] doing this for all the students we will get the highest marks.

the corresponding record in the student file will give us all the details about the topper. & hence we would cout the name, roll no and maxmarks.

using, cout<<"topper of division "<<userinputfordiv[6]<<" is "<<array.Name<<" whose roll no. is "<<array.Roll\_no<<" and the highest marks in this particular course are "<<maxmarks<<endl;

so the querie of the user has been correctly answered.

**(query 4)PROGRAM FOR HOSTELWISE DEPARTMENT STRENGTH**

We have a binary file in which student info is present. There is a structure in program containing all the details of thestudent.

The name of the structure is studentinfo. The use of the array hostel\_dept[20] is made(studentinfo hostel\_dept[20]).

hostel\_dept[i].name contains student's name, hostel\_dept[i].hostel\_number contains students' hostel number and hostel\_dept[i].dept\_name contains students' department name.Because the binary file is taken then we cannot access data by setting or shifting the pointer, there is a 'var' variable which load whole line of only one student info. After storing all data of a line, now we can access it.

.

I assigned a pointer (\*fp) using condition "while(!feof(fp))". pointer will sequentially read each line of the file by this command

''fread(&studentinfo1, sizeof(studentinfo), 1,fp)''. Intialise the integers 'row' , 'coloumn' , 'a' , i ,j to 0. An int array deptdata[5][5] is madefor displaying departments and hostel

Initially all array elements are 0. We use:

for(i=0, i<5, i++) {

for(j=0, j<5, j++){ (dept\_[i][j]=0;) } }

locating a row for each deppartment:

if(hostel\_dept.dept\_name=="computer science")

{ row=0;}

if(hostel\_dept.dept\_name=="electrical")

{ row=1;}

if(hostel\_dept.dept\_name=="chemical")

{ row=2;}

if(hostel\_dept.dept\_name=="mechanical")

{ row=3;}

if(hostel\_dept.dept\_name=="civil")

{ row=4;}

locating a coloum for each hostel:

By using switch condition we can determine coloumn 'a'

For the cases a=1,2,3,4 coloumn=a-1

and for case a=10 coloumn=4.

Everytime on reading a line from 'var' variable increment will be done in that deptdata[row][coloumn] by 1.

output is in the following manner:

cout<<"hostel number 1 2 3 4 10"<<endl;

cout<<"computer science ";

for(i=0,i<5,i++){

cout<<" "<<deptdata[0][i]<<" "<<endl;

}

cout<<"electrical ";

for(i=0,i<5,i++){

cout<<" "<<deptdata[1][i]<<" "<<endl;

}

cout<<"chemical ";

for(i=0,i<5,i++){

cout<<" "<<deptdata[2][i]<<" "<<endl;

}

cout<<"mechanical ";

for(i=0,i<5,i++){

cout<<" "<<deptdata[3][i]<<" "<<endl;

}

cout<<"civil ";

for(i=0,i<5,i++){

cout<<" "<<deptdata[4][i]<<" "<<endl;

}

}

By this the user will get the department wise strengths of hostels 1, 2, 3, 4,

10.

***(Query 6)PROGRAM FOR ADDITIONAL COURSES***

Those courses are namely, N.C.C, N.S.O & N.S.S.

Firstly a function(i.e; void additional\_courses(arguments)) is declared.

Start it by an opening curly brace ('{'). .

Declaring the file pointer(i.e; FILE \*infile; ).

Now, 'infile' is the pointer pointing to the base address of our binary stored file (Input file).

Linking pointer 'infile' to the stored file.

'filename' is a variable having the real stored file name in it.

It will be assigned some name in the main function.

Now, under a loop(i.e; while) having conditional expression( i.e;( !feof(infile) ) ), following things are to be done.

In a variable 'var', extract one block of data for one student's information from our stored file (Input file).

Simple "rb" command for opening in reading mode will be used.

Now, comparing the 'addi\_course' part of that student with N.C.C , N.S.O & N.S.S , implemented as follows :-

Here , this can be implemented in two ways , as follows :-

( 1.) Using one dimensional array as done below.

( 2.) Using two dimensional array as done below.

(1.)

Under this function, an array of the type "int" (i.e; int Number[3];) is declared.

All the three indices of the array "Number" (i.e; 0,1 & 2) are assigned to be zero, initially

First check will compare index "2" of the character array (i.e; char addi\_course[4] ) declared in the structure "student", with 'C'.

One 'if' condition will be used, and will be written as follows:-

(var.addi\_course[2]=='C').

If this condition here becomes true, then Number[0] will be increased by one. (i.e; Number[0]++ )

Second check will compare index "2" of the character array (i.e; char addi\_course[4] ) mentioned above, with 'O'.

If this condition becomes true, then Number[1] will be increased by one. (i.e; Number[1]++ )

Third check will compare the index "2" of the character array (i.e; char addi\_course[4] ) mentioned above , with 'S'.

If this condition is true , then Number[2] will be increased by one. (i.e; Number[2]++ )

This is how , as the loop will continue proceeding one by one over every student's data the respective counts will increase in the array 'Number'.

In the array 'Number', the members elements are assigned as follows:-

Number[0] will have the count for students in N.C.C.

Number[1] will have the count for students in N.S.O.

Number[2] will have the count for students in N.S.S.

(2.)

A two dimensinal array is declared first of the type "int" (i.e; Number[3][3]; ) along with three variables of type "int" as ncc,nso and nss.

The nine elements of the array 'Number' , are initialised so as to store 'N','C','C','N','S','O','N','S' & 'S' one by one in this order.

Now , complete 'var.addi\_course' can be compared with Number[0] , Number[1] & Number[2] using 'if' condition.

Only Number[0] etc; is written because it will just point to three different 'N'`s of N.C.C , N.S.O & N.S.S.

According as ncc,nso and nss variables are increased for respective matching in array 'Number'.

'ncc' will have the count for students in N.C.C.

'nso' will have the count for students in N.S.O.

'nss' will have the count for students in N.S.S.

Here as the loop continues, the position pointer reads over one by one student's data storage automatically. So, no need to reset the position of it.

Now by simple "cout" statements, number of students in N.C.C , N.S.O & N.S.S are printed.

At the end write 'return' as the return type is 'void'.

End the function by a closing curly brace ('}').

DISPLAY MENU

PROGRAM FOR STUDENT FILE DISPLAY

**INPUT**:  
              The administrator will ask to display the details of all the students.  
        **OUTPUT:**  
              Output will be the details mentioned below .  
         
      **DOCUMENTATION OF THE FIRST PART**  
1. Firstly I will make a window of size 20cm\*20 cm and further dividing it into 7 columns and 21 rows.In columns the following details will appear:  
        a) Name    b)Roll no   c)branch  d)division   e)home state   f)extra activity     g) Blood group  
2.All the rows will contain these datas for different students.          
         
    **HEADER FILES USED**   
1.<cstdio.h>   
2.<cassert>  
3.<ezwin.h>  
4.<cstring.h>  
  
  **FUNCTIONS USED**  
1.To make a window with the particular name and size we use **SimpleWindow MyWindow**("untitled ",20.0,20.0,Position(1.0 ,1.0 ))  
2.To open the window we will use **MyWindow.Open()** inside ApiMain and to check whether the window has opened or not we use**assert** command.  
3.we use **RenderRectangle** function with proper syntax to make a rectangle.  
4.We use **RenderLine** to draw a line using its initial and final position.  
5.we use **RenderText** to write into the respective blocks.  
6.We will also be using '**for**' loop for lines drawn equidistant.  
    
               
   **STRUCTURE OF STUDENT INFORMATION**         struct student\_info {  
             char Name[25];  
             char Roll\_no[12];  
             char Branch[22];  
             char Home\_state[22];  
             char extra\_act[5];            
             char blood\_group[5];  
  
  
**DOCUMENTATION OF THE SECOND PART**  
  
1.In file handling part first we have construct a structure with required attributes and then we are declaring a FILE pointer say 'infile' and associating it with the given file.  
2.then we will declare the int type recsize i.e. size of the struct student\_info.  
3.now I will open the binary file in reading mode and point the pointer to it with fopen command.  
                                              infile = fopen("filename","rb");  
  
4.now using 'fseek' command we can set the position of pointer. But it is always better to set its position to zero.  
5.now using 'fread' statement I will copy the file record into my structure using a while loop saying "while(!feof)".  
                                               fread(struct student\_info,recsize,1,infile);  
6.As the pointer was at the zero position it will copy the data till recsize into the structure.  
7.Now as all the data of the file has been transfered to the struct we will now start a 'for' loop to print the data student wise in the table.  
                          for(int i=0;i<21;i++){  
                          RenderText(Position(),Position(),[var.name](http://var.name/),colour)  
      similarly for roll no.,branch,etc..  and the variable i will be mentioned in the position part.  
  
8.Then we will display the whole information about student to the administration.

PROGRAM FOR COURSE FILE DISPLAY

Header files included:

#include"ezwin.h"

#include<cassert>

#include<cstring>

#include<cstdio>

functions included :

1 : For defining Ezwindows I will use function SimpleWindow() and by using

assert(MyWindow.GetStatus()==Window Open) I willcheck whether window is opened or not.

2 : For displaying the course in administrator part I am drawing a table which shows courses,instructor

for course and it's credit for particular division.For drawing table I am using Render.line and

Render.rectangle function.

3 : For writing a text which include course information I am using Render.text function.

4 : I am creating a data type Course by using stucture which includes division of students(int type) and a substructure which includes the data course name ,instructor name and credits for each course for different courses.

5 : Defining a variable var1 of data type course for holding the information about particular division.

6 : Creating a function administrator\_ login\_course\_info which includes the while loop described below.

7 : Using infile I am pointing the input file "course.dot".

8 : Using fread() function which reads the data of file and storing it in variable var1 using for different division by while loop.

9 : Termination condition for while loop is coded by feof().

10 : Using an array for storing the data for different course.

11 : Using a for loop for writing different text in different box.

12 : For this defining a character array includes course name,instructor name, credit in it.

13 : It will be done by strcpy which firstly copy the instructor name in array.

14 : After that course name and credits copying in array is doing by strcat whose work is catenation the data type in one array and display each variable in each box below of another.

15 : For converting the credit(int) into string so it will be copied on array I am converting the value of credit in to ASCII

code by defining a char array variable which holds the ASCCI value of credit for different coures.

16 : For printing the text in blocks I will use Render.text function .The position of blocks will keeps on changing depending upon

value of variable used in for loop and using character array I will print the text in blocks.

17 : Same type of function student\_login\_course\_info will be created for displaying the information for student login.

18 : First taking the value of roll no. from input of user login (global variable) and maching it with his own division by fseek().

19 : Maching will be done by reading course file by fread() and checking the student division with var1.division.when it will match

print the in formation of course by the coder similar to administrator\_ login\_course\_info function.

20 : Displaying of student part will same as that of administrator part.

                                                         

MARKS SHEET DISPLAY

**TEXTUAL DESCRIPTION OF THE PORTION ALLOCATED**  
       Our team handles the "QUERIES" portion of the project.In all my team is supposed to write six programs out of which i have taken one.This            is a brief description of what i have in my mind regarding this program.  
                 **DOCUMENTATION OF THE PROGRAM**  
                 I have to make roll no. and password panel for the starting window of our program.      
               **INPUT**  
                 the user will enter his/her roll no. and password to see his or her database.  
                 **OUTPUT**  
                 the moment user enters his/her roll no. a search will begin inside the index file which is binary file data that stores roll no. along with the associated record no.. if the roll no. matches with the record no. a search will begin inside the record file which contains students record along with the associated password. if the password matches with the queried password then screen will display the queried record of student.  
  
if anyone of the roll no. and password or both outmatches any of the details in index file and record file respectively then screen will display following message:  
 "INCORRECT ROLL NO. OR PASSWORD"   
               
  
  
                                                    **OUTLINE OF THE PROGRAM**  
1. I've been given 2 files in the binary format. one is the index.dat and the other is  the record.dat.  
  
2. index file contains roll no. along with with the associated record no. of the student and record file contains record along with the associated password.  
  
3.both of these binary files will be referred by their respective pointers (say p1 & p2). also two arrays will be made for roll no. and password: one named roll[] and other named password[]. size of each of these array will be same as the no. of students.    
  
4. first I'll ask user to enter his/her roll no. the moment user enters roll no. fopen will open index file by the following command:  
                  p1=fopen("index.dat","r")  
next inside the index file a loop will begin that will search the queried roll no. using following command:  
                  for(i=0,i<5000,i++){          //assumption:no of students are 5000//  
                  fread( );  
                  }  
if the queried roll no. matches with any of the roll no. in index file following output will be give to user;   
                 cout<<"enter your password:"<<endl; cin>>password[i];and compiler will move to record file for the next query.  
  
5. the moment user enters password compiler will search in the record file by following command.  
                for(i=o,i<5000,i++){  
                  
                }  
if the queried password matches with any of the password in the record file then the corresponding record file will apear by this command:  
               if (p2==password[i]){  
               break;  
               cout<<"result for your query:"<<endl;  
              }  
if any one of the roll no. and password or both outmatches those which are in respective index and record file the following command will be executed to show the message "incorrect roll no. or password"  
              else cout<<"incorrect roll no. or password"<<endl;

STUDENT PROCESSING MENU

1) ***Addition of new record to student file***

In our “Student Database”, there is a provision to add, modify, list and delete records , the operations that are imperative in any database management.

I am working on the modification part of the project.

Following is the report on the first stage progress of the work done by me for the project.

1) First of all, only an administrator should be allowed to modify any information related to the student.This will be ensured by checking the username and password entered in the beginning of the interface.

2)Then the user will be asked for the roll no. of the student whose records he/she wants to modify.The entered roll no. will first be verified.Then the index corresponding to the roll no. in the index\_file will be displayed.This will be done by calling the function get\_index\_of\_rollno().

3)The students information and marks information of the corresponding index will be wholly displayed.This will be done by calling the function display\_students\_info() and display\_students\_marks().

4)While modifying the record first we must ask the user which record/field he intends to modify i.e. from name,dob,home city, division, branch, hostel no. etc. This will be done using switch-break commands.

5) case 1-

      the field to be modified is in the students\_info file.(say name)

      The old name will be displayed and the user will be asked to enter the new name.

   case 2-

       the field to be modified is the marks of the student in a particular subject.

       The old marks will be displayed and the user will be asked to enter the new marks.The   course file will be opened and the credits for the course of the subject which has changed marks will be used in calculating the new SPI of the student.

    case 3-

       the field to be modified is the roll no. itself.

       Point to be noted here is that the user must know the old (unmodified) roll no. of the student beforehand. he will then be asked to enter the new roll no. the new roll no. has to be overwritten in the student\_info gfile as well as the index file.

after the completion of modification of record , the existing record will be overwritten by the new record in the appropriate file.

2***) Student data entry***  
Our team is been given the task of Student Data Processing.  
  
  
FUNCTIONS USED :-  Add\_student()  
WORK OF THE FUNCTION :- It will take the entry of new student from the administrator and store it in various corresponding files.  
OBJECTS USED :- student, Index,result  
  
ALGORITHM :-

1. The main task is to take entry of a new student.
2. Save the entered data in all the corresponding files namely student file, result file and index file.
3. All the data will be stored in an object of structure student\_info that contains all elements regarding students basic information.
4. The main challenge is to check the correctness of all the entries.
5. To check the validity of roll number, i will first count the total characters(can be integer or character),that should be equal to 9.
6. Later to check whether the character 'D' is placed in the right place or not.
7. Rest all elements should be integers in the roll number.
8. Next I will ask the roll number of the student.
9. The division of the student will be an integer.
10. The branch will be kept in an string(array of characters).
11. The entered branch will be checked against some pre-defined branches.
12. If the entered branch does not match, the entry will be discarded.
13. Next date of birth will be asked.
14. A function named check\_date will be called for checking the validity of the date.
15. If the date is found invalid, the entry will be discarded.
16. Hostel number and room number will be kept in short integer type variables.
17. Home state will be kept in a string.
18. The other activities(NSS/NCC/NSO) will be taken in another string and its validity will be checked .
19. If the entry is invalid, the entry will be discarded.
20. The blood group will be taken in a string.
21. Its validity will be checked against some pre-defined blood groups via a function named check\_blood\_group.
22. The password will be taken in a string.
23. The whole structure object will be written in the student file through the fwrite() function.
24. Parallel to these, an object of structure result will be created.
25. The roll number  entered by the user will be copied in it's roll number element.
26. All the elements of the array marks will initially be assinged 0.
27. The grade for extra\_acts will initially be "-" .
28. The SPI will be calculated by taking marks from the marks file and credits from the course file.
29. The calculated SPI will be stored in SPI element of the object.
30. The whole object will be written in the result file through the fwrite() function.
31. Parallel to this, roll number entered by the user will be copied in the corresponding element of the object of structure index.
32. This object will be written in the index file as above.

LIMITATIONS OF THE FUNCTIONS :-

1. It can't check the validity of the roll number fully.
2. It  can't store various group in the NSS/NSO courses.
3. It can not check the validity of the state.
4. The SPI calculated will not be accurate(will not be as per the actual procedure).
5. It cannot check the validity of the branch entered accurately.

***Marks processing menu***

***1)Marks modification and entry***

***FUNCTION NAME :- modify\_marks()  
  
VARIABLE USED :-    student,course,Index,result  
  
WORK OF THE FUNTION :- TO MODIFY OR ENTER THE MARKS OF STUDENT IN STUDENT DATABASE..............  
  
ALGORITHM :-***

1. 1st we will ask user the roll no...
2. then we will go to that index file and search at which position it is .
3. then go to that student file.
4. read record find the division
5. find courses in that division and display them.
6. show total courses with their no.
7. ask user which courses he will give ans as int
8. ex. 1 for chem
9. then open file result\_file of that coures
10. edit marks 1st all the marks will be 0.for course 1 marks will be entered in marks[0]
11. similarly for n marks [n-1]
12. and we will also calculate the spi
13. now what hapens that pointer shifts to next  record
14. so bring it back to that record again using fseek
15. and then rewrite it
16. and then ask user that if he wants to change the marks of other subject
17. if he says y
18. then same procedure will be followed..

***Limitations in my project***

No project can be flawless that too the one made in such a short span.

1. We are not using a strong concept of class. We could have done but explaining everyone the usage of that and implementing fluently in that way would have been much difficult and even if class was implemented then too every data member of that class had to be public that would disqualify usability of class.
2. Project is not much interactive , jumping between terminal to simple window can irritate the user using it.

……………………………………………………………………………………………………………………………………................

Refrences:

1. [www.cplusplus.com](http://www.cplusplus.com/)
2. Ez windows
3. Cohoon and Davidson