

```

/*Title: Library management system.
CBSE PROJECT 2017-2018
THE PROJECT IS PROGRAMED BY SWAKATH.S.U*/
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<string.h>
#include<stdio.h>
#include<process.h>
#include<ctype.h>
//globe variables
char date[10];

class Books
{
    //In bXYB suffix b represents book.
    char bname[30];
    char bcode[20];
    char bauthor[30];
    char btype[20];
    int btotal;
    int bstock;
    int bgiven;
    float bprice;
    int brating;
    struct bissue
    {
        char stuname[30];
        char sturollno[10];
        char issdate[10];
    };
    bissue bstu[10]; // for students who has taken this book

public:

    Books()//constructor
    {
        btotal=0;
        bstock=0;
        bprice=0;
        brating=0;
        bgiven=0;
    }

```

```

void binput();//function to input Book's data;
{
    cout<<"ENTER BOOK NAME : ";
    gets(bname);
    cout<<"ENTER BOOK CODE: ";
    gets(bcode);
    cout<<"ENTER AUTHOR NAME: ";
    gets(bauthor);
    cout<<"ENTER THE BOOK TYPE: ";
    gets(btype);
    cout<<"ENTER TOTAL COPIES: ";
    cin>>bttotal;
    cout<<"PRICE OF THE BOOK: ";
    cin>>bprice;
    cout<<"RATING OF THE BOOK: ";
    cin>>brating;
    cout<<"THANK YOU FOR THE INFORMATION";
    bstock=bttotal-bgiven;
}

void boutput(); //for displaying the datas
void blend(char nam[],char rol[],char isd[]);//for adding the data of the students to whom the
book is lend to
void breturn(char roll[]);

int ifbname(char name[]) //checks the give array with bname and returns 1 and 0 if false
{
    int s=0;
    if(strcmpi(bname,name)==0)
        s=1;
    return(s);
}

int ifbcode(char code[])
{
    int s=0;
    if(strcmp(bcode,code)==0)
        s=1;
    return(s);
}

char* retbname()
{
    char *ptr;

```

```

ptr=bname;
return ptr;
}

```

```

char* retbcode()
{
char *ptr;
ptr=bcode;
return ptr;
}
};

```

```

void Books::boutput()//function to output entire Book's data;
{
int i,j=1;
cout<<"BOOK NAME : "<<bname<<endl;
cout<<"BOOK CODE: "<<bcode<<endl;
cout<<"AUTHOR NAME: "<<bauthor<<endl;
cout<<"BOOK TYPE: "<<btype<<endl;
cout<<"TOTAL COPIES: "<<btotal<<endl;
cout<<"NUMBER OF COPIES IN STOCK: "<<bstock<<endl;
cout<<"NUMBER OF COPIES ISSUED: "<<bgiven<<endl;
cout<<"PRICE OF THE BOOK: "<<bprice<<endl;
cout<<"RATING OF THE BOOK: "<<brating<<endl;
cout<<"ISSUE TO:";
for(i=0;i<bgiven;++i)
{
cout<<j++<<"Name: "<<bstu[i].stuname<<" Rollno: "<<bstu[i].sturollno<<" ";
cout<<"Issued date:"<<bstu[i].issdate<<endl;
}
if(bgiven==0)
cout<<"NONE\n";
j=1;
}

```

```

void Books::blend(char nam[],char rol[],char isd[])//updates data memeber after leanding
{
bstock--;
strcpy(bstu[bgiven].stuname,nam);
strcpy(bstu[bgiven].sturollno,rol);
strcpy(bstu[bgiven].issdate,isd);
bgiven++;
}

```

```

void Books::breturn(char roll[])//updates data memebbers after returning
{
    int i,j;
    for(i=0;i<bgiven;++i)
    {
        if(strcmp(bstu[i].sturollno,roll)==0)
        {
            for(j=i;j<(bgiven-1);j++)
            {
                strcpy(bstu[j+1].stuname,bstu[j].stuname);
                strcpy(bstu[j+1].sturollno,bstu[j].sturollno);
                strcpy(bstu[j+1].issdate,bstu[j].issdate);
            }
            bgiven--;
            bstock++;
            cout<<"THE BOOK IS RETURNED THANK YOU";
            return ;
        }
    }
    cout<<"THE READER DID NOT TAKE THE ENTERED BOOK";
}

```

```

class member
{
    char memname[30]; //name of the member
    char memrollno[10]; //member's rollno
    int memtook;      //number of book taken
    struct tookbook
    {
        char tbname[30];
        char tbcode[20];
        char tbtdat[10];
    };
    tookbook took[10];
public:
    member()
    {memtook=0;}
    void meminput()
    {
        cout<<"ENTER THE NAME OF THE MEMBER: ";
        gets(memname);
        cout<<"ENTER THE ROLL NUMBER OF THE MEMBER: ";
    }
}

```

```

    gets(memrollno);
    cout<<"THE MEMBER HAS BEING ADDED INTO THE LABIRARY\n\n";
}
void memoutput();
int ifmemrollno(char rollno[])//gets a char value and checks it with memrollno and returns 1 if
true else 0
{
    int s=0;
    if(strcmp(memrollno,rollno)==0)
        s=1;
    return(s);
}

void memlend(char tsname[],char tscod[],char td[]);
void memret(char code[]);

char* retmname()
{
    char *ptr;
    ptr=memname;
    return ptr;
}

char* retmrollno()
{
    char *ptr;
    ptr=memrollno;
    return ptr;
}
};
void member::memoutput()
{
    int i;
    cout<<"MEMBER NAME: "<<memname<<endl;
    cout<<"MEMBER ROLL NUMBER: "<<memrollno<<endl;
    cout<<"NUMBER OF BOOKS TAKEN: "<<memtook<<endl;
    if(memtook!=0)
    {
        cout<<"DETAILS OF TAKEN BOOKS:\n";
        for(i=0;i<memtook;++i)
        {
            cout<<i+1<<" ) BOOK NAME: "<<took[i].tbname<<endl;
            cout<<" BOOK CODE: "<<took[i].tbcode<<endl;
        }
    }
}

```

```

    cout<<" DATE: "<<took[i].tbtdate<<endl;
}
}
}

```

```

void member::memret(char code[])
{
    int i,j;
    for(i=0;i<memtook;++i)
    {
        if(strcmp(took[i].tbcode,code)==0)
        {
            for(j=i;j<(memtook-1);++j)
            {
                strcpy(took[j+1].tbname,took[j].tbname);
                strcpy(took[j+1].tbcode,took[j].tbcode);
                strcpy(took[j+1].tbtdate,took[j].tbtdate);
            }
            memtook--;
            return;
        }
    }
    cout<<"THE ENTERED BOOK IS NOT TAKEN BY THE MEMBER \n\n";
}

```

```

void member::memlend(char tsname[],char tscode[],char td[])
{
    strcpy(took[memtook].tbname,tsname);
    strcpy(took[memtook].tbcode,tscode);
    strcpy(took[memtook].tbtdate,td);
    memtook++;
}

```

```

void main()
{
    int flg=0,ch;
    clrscr();
    ifstream f1("password.txt");
    int YES=0;
    char userI[30],passwordI[30],userC[30],passwordC[30];
    cout<<"\t\tLIBRARY MAINTAINANCE SYSTEM\n\n";
}

```

```

f1.getline(userC,80);
f1.getline(passwordC,80);
do
{
if(flag!=0)
{
do
{
cout<<"THE ENTERED USERNAME AND PASSWORD ARE WRONG PLEASE\n";
cout<<"ENTER 1 - TO TRY AGAIN or 2 - TO EXIT : ";
cin>>ch;
switch(ch)
{
case 1:break;
case 2:cout<<"  THANK YOU\n";
exit(0);
default:cout<<"Enter valid option\n";
}
}
while(ch!=1&&ch!=2);
}
cout<<"ENTER USER NAME: ";
gets(userI);
cout<<"ENTER PASSWORD: ";
gets(passwordI);
if(strcmp(userI,userC)==0&&strcmp(passwordI,passwordC)==0)
{
cout<<"\n\t\tWELCOME\n";
YES++;
}
else
flag++;
}
while(YES==0);
cout<<"ENTER TODAYS DATE : ";
gets(date);
cout<<endl;
void MAIN_MENU();
MAIN_MENU();
getch();
}

```

```

void SEARCHBOOK()
{
    repeat:
    Books ser;
    ifstream fb("BOOKS.DAT",ios::binary);
    if(!fb)
    {
        cout<<"ERROR";
        return;
    }
    char code[10];
    int ch,flg=0;
    flg=0;
    cout<<"ENTER THE BOOK ID FOR SEARCH : ";
    gets(code);
    while(fb.read((char*)&ser,sizeof(ser)))
    {
        if(ser.ifbcode(code)==1)
        {
            cout<<"THE ENTERED BOOK'S INFORMATION\n";
            ser.boutput();
            flg=1;
            break;
        }
    }
    if(flg==0)
    cout<<"THE ENTERED BOOK IS NOT FOUND IN DATA BASE";
    fb.close();
    int r;
    option:
    cout<<"\n\nCHOOSE YOUR OPTION:\n1.CONTINUE WITH BOOK SEARCH\n2.GO TO MAIN MENU\n";
    cin>>r;
    switch(r)
    {
        case 1: goto repeat;
        case 2: break;
        default:cout<<"INVALID OPTION PLEASE TRY AGAIN";
                goto option;
    }
}

```



```

void SEARCHMEMBER()
{
    repeat:
    char rollno[10];
    int flg=0;
    ifstream fm("MEMBER.DAT",ios::binary);
    if(!fm)
    {
        cout<<"ERROR\n";
        return;
    }
    flg=0;
    cout<<"Enter the user roll number : ";
    gets(rollno);
    member mem;
    while(fm.read((char*)&mem,sizeof(mem)))
    {
        if(mem.ifmemrollno(rollno)==1)
        {
            mem.memoutput();
            flg=1;
            break;
        }
    }
    if(flg==0)
    cout<<"THE MEMBER WITH THE GIVEN ROLL NUMBER IS NOT FOUND IN THE DATA
BASE";
    fm.close();
    int r;
    option:
    cout<<"\n\nCHOOSE YOUR OPTION:\n1.CONTINUE WITH MEMBER SEARCH\n2.GO BACK
TO MAIN MENU\n";
    cin>>r;
    switch(r)
    {
        case 1: goto repeat;
        case 2: break;
        case 3:cout<<"INVALID OPTION PLEASE TRY AGAIN";
            goto option;
    }
}

```

```

void LENDING()

```

```

{
    repeat:
    char bcode[10],rollno[10],bname[30],mname[30];
    int flg=0;
    fstream fm("MEMBER.DAT",ios::in|ios::out|ios::binary);
    fstream fb("BOOKS.DAT",ios::in|ios::out|ios::binary);
    if(!fm||!fb)
    {
        cout<<"ERROR\n";
        return;
    }
    member mem;
    Books bok;
    cout<<"ENTER THE ROLL NUMBER OF THE MEMBER: ";
    gets(rollno);
    cout<<"ENTER THE BOOK ID: ";
    gets(bcode);
    while(fm.read((char*)&mem,sizeof(mem)))
    {
        if(mem.ifmemrollno(rollno)==1)
        {
            strcpy(mname,mem.retname());
            flg=1;
            break;
        }
    }
    if(flg==0)
    {
        cout<<"ROLL NUMBER NOT FOUND";
        fb.close();
        fm.close();
        goto option;
    }
    flg=0;
    while(fb.read((char*)&bok,sizeof(bok)))
    {
        if(bok.ifbcode(bcode)==1)
        {
            strcpy(bname,bok.retname());
            flg=1;
            break;
        }
    }
}

```

```

if(flag==0)
{
cout<<"BOOK NOT FOUND";
fb.close();
fm.close();
goto option;
}
bok.blend(mname,rollno,date);
mem.memlend(bname,bcode,date);
int n;
n=(-sizeof(bok));
fb.seekp(n,ios::cur);
fb.write((char*)&bok,sizeof(bok));
n=(-sizeof(mem));
fm.seekp(n,ios::cur);
fm.write((char*)&mem,sizeof(mem));
cout<<"THE DATAS IS UPDATED AFTER LENDING";
int r;
fb.close();
fm.close();
option:
cout<<"\n\nENTER YOUR OPTION\n1.CONTINUE WITH LENDING\n2.GO TO MAIN
MENU\n";
cin>>r;
switch(r)
{
case 1: goto repeat;
case 2: break;
default:cout<<"INVALID OPTION PLEASE TRY AGAIN";
goto option;
}
}

void RETURNING()
{
repeat:
fstream fm("MEMBER.DAT",ios::in|ios::out|ios::binary);
fstream fb("BOOKS.DAT",ios::in|ios::out|ios::binary);
if(!fm||!fb)
{
cout<<"ERROR\n";
return;
}
}

```

```

char rollno[10],bcode[10];
int n,flg=0;
cout<<"ENTER THE ROLL NUMBER OF THE MEMBER: ";
gets(rollno);
cout<<"ENTER THE RETURNING BOOK CODE: ";
gets(bcode);
member mem;
Books bok;
while(fb.read((char*)&bok,sizeof(bok)))
{
    if(bok.ifbcode(bcode)==1)
    {
        flg=1;
        break;
    }
}
if(flg==0)
{
    cout<<"THE ENTERED BOOK IS NOT FOUND IN THE DATA BASE";
    fb.close();
    fm.close();
    goto option;
}
flg=0;
while(fm.read((char*)&mem,sizeof(mem)))
{
    if(mem.ifmemrollno(rollno)==1)
    {
        flg=1;
        break;
    }
}
if(flg==0)
{
    cout<<"ROLL NUMBER NOT FOUND";
    fb.close();
    fm.close();
    goto option;
}
bok.breturn(rollno);
mem.memret(bcode);
n=(-sizeof(bok));
fb.seekp(n,ios::cur);

```

```

fb.write((char*)&bok,sizeof(bok));
n=(-sizeof(mem));
fm.seekp(n,ios::cur);
fm.write((char*)&mem,sizeof(mem));
cout<<"THE DATAS ARE UPDATED AFTER RETURNNING";
int r;
fb.close();
fm.close();
option:
cout<<"\n\nCHOOSE YOUR OPTION\n1.CONTINUE WITH RETURNING\n2.GO TO MAIN
MENU\n";
cin>>r;
switch(r)
{
case 1: goto repeat;
case 2: break;
default:cout<<"INVALID OPTION PLEASE TRY AGAIN";
        goto option;
}
}

```

```

void ADDBOOK()
{
repeat:
Books bok;
fstream fb("BOOKS.DAT",ios::binary|ios::in|ios::out);
if(!fb)
{
cout<<"ERROR\n";
return;
}
bok.binput();
fb.write((char*)&bok,sizeof(bok));
int r;
fb.close();
option:
cout<<"\n\nCHOOSE YOU OPTION\n1.CONTINUE TO ADD BOOKS\n2.GO TO MAIN
MENU\n";
cin>>r;
switch(r)
{
case 1: goto repeat;
case 2: break;

```

```

        default: cout<<"INVALID OPTION PLEASE TRY AGAIN";
                goto option;
    }
}

void ADDMEMBER()
{
    repeat:
    member mem;
    ofstream fm("MEMBER.DAT",ios::binary|ios::app);
    if(!fm)
    {
        cout<<"ERROR\n";
        return;
    }
    mem.meminput();
    fm.write((char*)&mem,sizeof(mem));
    int r;
    fm.close();
    option:
    cout<<"\n\nCHOOSE YOUR OPTION\n1.CONTINUE TO ADD BOOKS\n2.GO TO MAIN
MENU\n";
    cin>>r;
    switch(r)
    {
        case 1: goto repeat;
        case 2: break;
        default:cout<<"INVALID OPTION PLEASE TRY AGAIN";
                goto option;
    }
}

void GROUPDETAILS()
{
    repeat:
    ifstream fb("BOOKS.DAT",ios::binary);
    ifstream fm("MEMBER.DAT",ios::binary);
    if(!fb||!fm)
    {
        cout<<"ERROR\n";
        return;
    }
    int ch,i=1;

```

```

char bcode[10],rollno[10],mname[30],bname[30];
member mem;
Books bok;
cout<<"CHOOSE YOUR OPTION\n";
cout<<"1.BOOK DETAILS\n2.MEMBER DETAILS\n";
cin>>ch;
switch(ch)
{
case 1 : while(fb.read((char*)&bok,sizeof(bok)))
        {
            strcpy(bname,bok.retbyname());
            strcpy(bcode,bok.retbycode());
            cout<<i++<<" ) BOOK NAME: "<<bname<<endl<<" BOOK CODE: "<<bcode<<endl;
        }
        i=1;
        cout<<endl;
        break;
case 2 : while(fm.read((char*)&mem,sizeof(mem)))
        {
            strcpy(rollno,mem.retrollno());
            strcpy(mname,mem.retbyname());
            cout<<i++<<" ) MEMBER NAME: "<<mname<<endl<<" MEMBER ROLL NUMBER:
"<<rollno<<endl;
        }
        i=1;
        cout<<endl;
        break;
default :cout<<"INVAILD OPTION";
        fb.close();
        fm.close();
        goto option;
}
int r;
fm.close();
fb.close();
option:
cout<<"\n\nCHOOSE YOUR OPTION\n1.CONTINUE WITH GROUP SEARCH\n2.GO TO
MAIN MENU\n";
cin>>r;
switch(r)
{
case 1: goto repeat;
case 2: break;

```

```

        default:cout<<"INVALID OPTION PLEASE TRY AGAIN";
                goto option;
    }
}

void DELETEFILE()
{
    repeat:
    ifstream fb("BOOKS.DAT",ios::binary);
    ifstream fm("MEMBER.DAT",ios::binary);
    ofstream tb("TEMPBOOK.DAT",ios::binary);
    ofstream tm("TEMPMEM.DAT",ios::binary);
    if(!fm||!fb||!tb||!tm)
    {
        cout<<"ERROR\n";
        return;
    }
    Books bok;
    member mem;
    char bcode[10],rollno[10];
    int ch,flg=0;
    cout<<"CHOOSE YOUR OPTION\n";
    cout<<"1.DELETE A MEMBER\n2.DELETE A BOOK\n";
    cin>>ch;
    switch(ch)
    {
        case 1 : cout<<"ENTER THE ROLL NUMBER OF THE MEMBER : ";
                gets(rollno);
                while(fm.read((char*)&mem,sizeof(mem)))
                {
                    if(mem.ifmemrollno(rollno)==1)
                    flg=1;
                    else
                    tm.write((char*)&mem,sizeof(mem));
                }
                if(flg==0)
                {
                    cout<<" A MEMBER WITH THE ENTERED ROLL NUMBER IS NOT FOUND IN THE
DATA BASE";
                    fm.close();
                    fb.close();
                    tm.close();
                    tb.close();
                }
            }
    }
}

```



```

        remove("TEMPMEM.DAT");
        remove("TEMPBOOK.DAT");
        goto option;
    }
    else
        cout<<"THE MEMBER HAS BEEN REMOVED FROM THE DATA BASE";
        fm.close();
        fb.close();
        tm.close();
        tb.close();
        remove("MEMBER.DAT");
        rename("TEMPMEM.DAT","MEMBER.DAT");
        remove("TEMPBOOK.DAT");
        break;
case 2 : cout<<"ENTER THE BOOK CODE : ";
        gets(bcode);
        while(fb.read((char*)&bok,sizeof(bok)))
        {
            if(bok.ifbcode(bcode)==1)
                flg=1;
            else
                tb.write((char*)&bok,sizeof(bok));
        }
        if(flg==0)
        {
            cout<<" BOOK WITH THE ENTERED BOOK CODE IS NOT FOUND IN THE DATA
BASE";
            fm.close();
            fb.close();
            tm.close();
            tb.close();
            remove("TEMPMEM.DAT");
            remove("TEMPBOOK.DAT");
            goto option;
        }
        else
            cout<<"THE BOOK HAS BEEN REMOVED FORM THE DATA BASE";
            fb.close();
            fm.close();
            tb.close();
            tm.close();
            remove("BOOKS.DAT");
            rename("TEMPBOOK.DAT","BOOKS.DAT");

```

```

        remove("TEMPMEM.DAT");
        break;
default :cout<<"INVALID OPTION";
        fm.close();
        fb.close();
        tm.close();
        tb.close();
        remove("TEMPMEM.DAT");
        remove("TEMPBOOK.DAT");
        goto option;

}
int r;
option:
cout<<"\n\nENTER YOUR OPTION\n1.CONTINUE WITH DELETION\n2.GO TO MAIN
MENU\n";
cin>>r;
switch(r)
{
    case 1: goto repeat;
    case 2: break;
    default:cout<<"INVALID OPTION PLEASE TRY AGAIN";
            goto option;
}
}

```

```

void MAIN_MENU()
{
    int choice;
    do
    {
        cout<<"\n\nMAIN MENU:\n";
        cout<<"Select a option to process\n";
        cout<<"1.SEARCH INDIVIDUAL BOOK \n";
        cout<<"2.SEARCH INDIVIDUAL LIBRARY MEMBER \n";
        cout<<"3.LENDING\n";
        cout<<"4.RETURNING\n";
        cout<<"5.ADD NEW BOOK\n";
        cout<<"6.CREATE NEW MEMBER\n";
        cout<<"7.DISPLAY GROUP DETAILS\n";
        cout<<"8.DELETE FILES\n";
        cout<<"9.EXIT\n";
    }
}

```

```
cin>>choice;
switch(choice)
{
    case 1: SEARCHBOOK();
            break;
    case 2: SEARCHMEMBER();
            break;
    case 3: LENDING();
            break;
    case 4: RETURNING();
            break;
    case 5: ADDBOOK();
            break;
    case 6: ADDMEMBER();
            break;
    case 7: GROUPDETAILS();
            break;
    case 8: DELETEFILE();
            break;
    case 9: break;
    default :cout<<"\tPLEASE ENTER VALID CHOICE\n\n";
}
}
while(choice!=9);
}
```