<http://www.bitavoncpp.com/hospital-management-system-in-c-project.html>

include<iostream.h> //for input and output stream regulation

#include<process.h> //for exit()

#include<string.h> //for strlen() and strcmp()

#include<conio.h> //for getch() and clrscr()  
#include<stdio.h>

class all //declaration for class "all"  
{  
private:  
struct address  
{  
int house;  
char city[30];  
char dist[30];  
char state[30];  
char street[30];  
char country[30];  
};  
struct age  
{  
int day;  
int month;  
int year;  
};  
struct patient\_info  
{  
age A1; //nested structure inplemented  
address AD1; //nested structure implemented  
int sex;  
int reg\_no;  
int bld\_group;  
char name[50];  
int martial\_status;  
}PI[100];  
int task;  
protected: //functions declared  
void search\_menu();  
void search\_city();  
void exit\_function();  
void search\_show\_info();  
void search\_blood\_group();  
void enter\_patient\_info();  
void show\_patient\_detail();  
void after\_search\_options();  
void after\_restore();  
public:  
void tasks();  
void recycle\_bin();  
void delete\_entry();  
void billing();  
void after\_delete\_options();  
int s\_group;  
int s\_choice;  
int en\_del\_index;  
int delete\_choice;  
char ch;  
char answer;  
char answer1;  
char s\_city[30];  
char exit\_answer;  
char delete\_confirm;  
char after\_search\_answer;  
}; //end of class "all"

class date //declaration for class "date"  
{  
private:  
int date;  
int month;  
int year;  
public:  
void enter\_date();  
void show\_date();  
}; //end of class "date"

class dob //declaration for class "dob"  
{  
private:  
struct dob1  
{  
int date;  
int month;  
int year;  
int rem;  
}DOB11[100];  
public:  
char birth\_answer;  
void show\_date();  
void enter\_date();  
void search\_show\_date();  
}; //end of class "dob"

class temp //declaration for class "temp"  
{  
public:  
int m; //temporary variables declared with global scope  
int i;  
int j;  
int k;  
int d;  
int e;  
int f;  
int rem;  
int temp;  
int count;  
int regis;  
int index;  
int entry;  
int serial;  
int attempt;  
int current;  
int d\_index;  
int ssi\_count;  
int show\_count;  
int delete\_show;  
int search\_index;  
int search\_count;  
int current\_year;  
int delete\_count;  
int search\_number;  
int restore\_serial;  
int delete\_attempt;  
int restore\_attempt;  
int entry\_index[100];  
int after\_search\_choice;  
int after\_restore\_choice;  
char enter\_now;  
char restore\_confirm;  
char duplicate\_answer;  
char delete\_all\_confirm;  
char restore\_all\_confirm;  
char after\_search\_answer;  
temp() //constructor for temp invoked  
{  
i=0;  
j=0;  
d=0;  
e=0;  
f=0;  
serial=0;  
current=0;  
d\_index=0;  
ssi\_count=0;  
show\_count=0;  
delete\_show=0;  
delete\_count=0;  
delete\_attempt=0;  
restore\_attempt=0;  
}; //end of constructor for temp

//destructor for temp invoked  
}; //end of class "temp"

all A1; //object for class "all" declared  
date D1; //object for class "date" declared  
dob DOB1; //object for class "dob" declared  
temp T1; //object for class "temp" declared

void main() //main function  
{  
T1.count=0;  
for(T1.m=1;T1.m<=100;T1.m++) //default index value set to -1 for each entry  
{  
T1.entry\_index[T1.m]=-1;  
}  
cout<<"Welcome to..."<<"\n";  
cout<<"\n\n\t\t\t\*\*\*HOSPITAL MANAGEMENT SOFTWARE\*\*\*"<<"\n";  
cout<<"\n\t\t\t By Radhika Shukla, Shraddha Sarna, Nikita Kumar "<<"\n"; D1.enter\_date();  
A1.tasks();  
} //end of main function

void all::tasks()  
{  
clrscr();  
T1.attempt=0;  
T1.d\_index=0;  
T1.delete\_attempt=0;  
D1.show\_date();  
cout<<"\n\n\t\t\t\*\*\*HOSPITAL MANAGEMENT SOFTWARE\*\*\*"<<"\n";  
cout<<"\t\t\t By Radhika Shukla, Shraddha Sarna, Nikita Kumar "<<"\n";  
cout<<"\t\t\t Guided By Ramesh Singh Parihar";  
cout<<"\n\n\t\t\t \*\*\*Hospital Management Tasks\*\*\*"<<"\n";  
cout<<"\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<"\n";  
cout<<"\n\n\tPlease select a task to do...."<<"\n";  
cout<<"\n\n\t1. Enter a new patient information "<<"\n";  
cout<<"\t2. View detail of existing patient "<<"\n";  
cout<<"\t3. Search by city or blood group"<<"\n";  
cout<<"\t4. Delete entry "<<"\n";  
cout<<"\t5. Recycle Bin "<<"\n";  
cout<<"\t6. Prepair Bill "<<"\n";  
cout<<"\t7. Exit "<<"\n\n\n";  
cout<<"\a Enter your task serial : ";  
cin>>task;  
switch(task)  
{  
case 1:{  
A1.enter\_patient\_info();  
break;  
}  
case 2:{  
A1.show\_patient\_detail();  
break;  
}  
case 3:{  
A1.search\_menu();  
break;  
}  
case 4:{  
A1.delete\_entry();  
break;  
}  
case 5:{  
A1.recycle\_bin();  
break;  
}  
case 6:{  
A1.billing();  
break;  
}  
case 7:{  
A1.exit\_function();  
break;  
}  
default:{  
clrscr();  
cout<<"\n\n\n\n\n\tInvalid task serial !!!"<<"\n";  
cout<<"\tValid options are 1-6 !!!"<<"\n";  
cout<<"\tPress any key to return to the main task menu....\a"<<"\n";  
getch();  
clrscr();  
A1.tasks();  
}  
} //end of switch for task  
} //end of function

void date::enter\_date()  
{  
cout<<"\n\nFirst of all I need the current date ..."<<"\n";  
cout<<"\nYear : ";  
cin>>year;  
while(year<2000||year>10000)  
{  
clreol();  
cout<<"\aInvalid input for year !!!"<<"\n";  
clreol();  
cout<<"Please enter the year correctly : "<<"\n";  
cin>>year;  
}  
T1.current\_year=year;  
cout<<"Month : ";  
clreol();  
cin>>month;  
while(month<=0||month>12) //if month is greater then 12 or less then 1  
{  
clreol();  
cout<<"\nInvalid input for month !!!"<<"\n";  
clreol();  
cout<<"Please enter the month correctly : \a"<<"\n";  
clreol();  
cin>>month;  
}  
cout<<"Date : ";  
clreol();  
switch(month)  
{  
case 1:  
case 3:  
case 5:  
case 7:  
case 8:  
case 10:  
case 12:{  
cin>>date;  
while(date<1||date>31) //for months with 31 days  
{  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : "<<"\n";  
clreol();  
cin>>date;  
}  
break;  
}  
case 2:{  
cin>>date; //for february  
if(year%4==0)  
{  
while(date<0||date>29) //for leap year  
{  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : "<<"\n";  
clreol();  
cin>>date;  
}  
}  
else  
{  
while(date<0||date>28) //for non-leap year  
{  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : "<<"\n";  
clreol();  
cin>>date;  
}  
}  
break;  
}  
default:{ //for other months with 30 days  
cin>>date;  
while(date<1||date>30)  
{  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : "<<"\n";  
clreol();  
cin>>date;  
}  
break;  
}  
} //end of switch  
} //end of function

void date::show\_date()  
{  
clrscr();  
cout<<"Hello....\n It's ";  
cout<<date;  
T1.rem=date%10;  
switch(date)  
{  
case 11:  
case 12:  
case 13:  
case 14:  
case 15:  
case 16:  
case 17:  
case 18:  
case 19:  
case 20:{  
cout<<"th ";  
goto over;  
}  
}  
switch(T1.rem)  
{  
case 1:{  
cout<<"st ";  
break;  
}  
case 2:{  
cout<<"nd ";  
break;  
}  
case 3:{  
cout<<"rd ";  
break;  
}  
default:{  
cout<<"th ";  
break;  
}  
} //end of switch  
over: //label for jump statement  
switch(month)  
{  
case 1:{  
cout<<"January , ";  
break;  
}  
case 2:{  
cout<<"February , ";  
break;  
}  
case 3:{  
cout<<"March , ";  
break;  
}  
case 4:{  
cout<<"April , ";  
break;  
}  
case 5:{  
cout<<"May , ";  
break;  
}  
case 6:{  
cout<<"June , ";  
break;  
}  
case 7:{  
cout<<"July , ";  
break;  
}  
case 8:{  
cout<<"August , ";  
break;  
}  
case 9:{  
cout<<"September , ";  
break;  
}  
case 10:{  
cout<<"October , ";  
break;  
}

case 11:{  
cout<<"November , ";  
break;  
}  
case 12:{  
cout<<"December , ";  
break;  
}  
} //end of switch  
cout<<year<<" \n";  
}  
//end of function  
void all::enter\_patient\_info()  
{  
clrscr();  
if(T1.current==100) //to avoid overflow of data in the database  
{  
cout<<"\n\n\n\t Database is full !!! ";  
cout<<"\n\t Sorry, you cannot enter any more information !!! ";  
cout<<"\n\t Information of maximum 100 patients can be entered !!!! ";  
cout<<"\n\t Press any key to return to the main menu....\a ";  
getch();  
A1.tasks();  
}  
answer='y';  
if(T1.count==0)  
{  
T1.serial=1;  
}  
else  
{  
T1.i=T1.serial;  
}  
for(T1.i=T1.serial;answer=='y'||answer=='Y';T1.i++)  
{  
cout<<"\a";  
T1.current++;  
PI[T1.i].reg\_no=T1.i;  
T1.temp=T1.serial;  
clrscr();  
T1.entry\_index[T1.i]=1;  
cout<<"\n\n\t \*\*\*ENTERING INFORMATION FOR PATIENT SERIAL NUMBER "<<T1.i<<" \*\*\*"<<"\n";  
cin.get(ch);  
cout<<"\nRegistration Number : "<<PI[T1.i].reg\_no<<"\n\n";  
cout<<"Enter the name of patient : ";  
clreol();  
cin.getline(PI[T1.i].name,50);  
while(strlen(PI[T1.i].name)==0)  
{  
cout<<"\a\nPlease enter a name : "; //if enter is pressed before netering a name  
cin.getline(PI[T1.i].name,50);  
cout<<"\n";  
}  
for(T1.k=1;T1.k<T1.serial;T1.k++) //loop to avoid duplicate entries  
{  
if(strcmp(PI[T1.i].name,PI[T1.k].name)==0) //if match is found  
{  
if(T1.entry\_index[T1.k]==0) //if entry is already deleted  
{  
continue;  
}  
else //if entry still exists  
{  
T1.current-=1; //current no. of entries remains same  
clreol();  
cout<<"\n\n\n\tThe patient is already admitted in the hospital !!!";  
cout<<"\n\tWant to view info of this registration number in detail (Y/N) : \a";  
cin>>T1.duplicate\_answer;  
while(T1.duplicate\_answer!='Y'&&T1.duplicate\_answer!='y'&&T1.duplicate\_answer!='N'&&T1.duplicate\_answer!='n')  
{  
cout<<"\n\t\aPlease re enter a correct option (Y/N) : ";  
cin>>T1.duplicate\_answer;  
}  
if(T1.duplicate\_answer=='y'||T1.duplicate\_answer=='Y')  
{  
T1.entry\_index[T1.i]=-1;  
T1.regis=T1.k;  
T1.d\_index=1;  
A1.show\_patient\_detail();  
}  
else  
{  
A1.tasks();  
}  
}  
}  
} //end of "for loop" to prevent duplicate entries  
cout<<"\nSex ( 1-Male , 2-Female ) : ";  
clreol();  
cin>>PI[T1.i].sex;  
while(PI[T1.i].sex!=1&&PI[T1.i].sex!=2)  
{  
cout<<"\a\nInvalid input for sex of patient !!!";  
cout<<"\nSex : ";  
clreol();  
cin>>PI[T1.i].sex;  
}  
cout<<"\nMartial status( 1-Married , 2-Single ) : ";  
clreol();  
cin>>PI[T1.i].martial\_status;  
while(PI[T1.i].martial\_status<1||PI[T1.i].martial\_status>2)  
{  
cout<<"\a\nInvalid input for martial status !!!"<<"\n";  
cout<<"Enter a valid martial status : "<<"\n";  
clreol();  
cin>>PI[T1.i].martial\_status;  
}  
cin.get(ch);  
clreol();  
clreol();  
cout<<"\n \*\*\* Blood group \*\*\*"<<"\n";  
clreol();  
cout<<" 1. A+ "<<"\n";  
clreol();  
cout<<" 2. A- "<<"\n";  
clreol();  
cout<<" 3. B+ "<<"\n";  
clreol();  
cout<<" 4. B- "<<"\n";  
clreol();  
cout<<" 5. AB+ "<<"\n";  
clreol();  
cout<<" 6. AB- "<<"\n";  
clreol();  
cout<<" 7. O+ "<<"\n";  
clreol();  
cout<<" 8. O- "<<"\n";  
clreol();  
cout<<"\nEnter a blood group : ";  
clreol();  
cin>>PI[T1.i].bld\_group;  
switch(PI[T1.i].bld\_group)  
{  
case 1:  
case 2:  
case 3:  
case 4:  
case 5:  
case 6:  
case 7:  
case 8:{  
break;  
}  
default:{  
while(PI[T1.i].bld\_group!=1&&PI[T1.i].bld\_group!=2&&  
PI[T1.i].bld\_group!=3&&PI[T1.i].bld\_group!=4&&  
PI[T1.i].bld\_group!=5&&PI[T1.i].bld\_group!=6&&  
PI[T1.i].bld\_group!=7&&PI[T1.i].bld\_group!=8)  
{  
clreol();  
cout<<"\aInvalid input !!!"<<"\n";  
clreol();  
cout<<"Please enter a valid Blood Group : "<<"\n";  
clreol();  
cin>>PI[T1.i].bld\_group;  
} //end of while  
break;  
}  
} //end of switch  
cin.get(ch);  
clreol();  
cout<<"\n\n \*\*\*ENTERING ADDRESS\*\*"<<"\n\n";  
clreol();  
cout<<"House number : ";  
clreol();  
cin>>PI[T1.i].AD1.house;  
while(PI[T1.i].AD1.house<=0)  
{  
clreol();  
cout<<"\a\nInvalid input for house number !!!";  
clreol();  
cout<<"\nAgain enter the house number : ";  
clreol();  
cin>>PI[T1.i].AD1.house;  
}  
cin.get(ch);  
clreol();  
cout<<"Street : ";  
clreol();  
cin.getline(PI[T1.i].AD1.street,30);  
while(strlen(PI[T1.i].AD1.street)==0)  
{  
clreol();  
cout<<"\a\nPlease enter a street : "; //if enter is pressd before input for street  
clreol();  
cin.getline(PI[T1.i].AD1.street,50);  
cout<<"\n";  
}  
cout<<"City : ";  
clreol();  
cin.getline(PI[T1.i].AD1.city,30);  
while(strlen(PI[T1.i].AD1.city)==0)  
{  
clreol();  
cout<<"\a\nPlease enter a city : "; //if enter is pressd before input for city  
clreol();  
cin.getline(PI[T1.i].AD1.city,50);  
cout<<"\n";  
}  
cout<<"District : ";  
clreol();  
cin.getline(PI[T1.i].AD1.dist,30);  
while(strlen(PI[T1.i].AD1.dist)==0)  
{  
clreol();  
cout<<"\a\nPlease enter a district : "; //if enter is pressed before input for district  
clreol();  
cin.getline(PI[T1.i].AD1.dist,30);  
cout<<"\n";  
}  
cout<<"State : ";  
clreol();  
cin.getline(PI[T1.i].AD1.state,30);  
while(strlen(PI[T1.i].AD1.state)==0)  
{  
clreol();  
cout<<"\a\nPlease enter a state : "; //if enter is pressed before input for state  
clreol();  
cin.getline(PI[T1.i].AD1.state,50);  
cout<<"\n";  
}  
cout<<"Country : ";  
clreol();  
cin.getline(PI[T1.i].AD1.country,30);  
while(strlen(PI[T1.i].AD1.country)==0)  
{  
clreol();  
cout<<"\a\nPlease enter a country : "; //if enter is pressd before input for country  
clreol();  
cin.getline(PI[T1.i].AD1.country,50);  
cout<<"\n";  
}  
DOB1.enter\_date();  
cin.get(ch);  
clreol();  
clreol();  
cout<<"\n\nWant to enter information for another patient (Y/N) ? ";  
clreol();  
cin>>answer;  
while(answer!='Y'&&answer!='y'&&answer!='N'&&answer!='n')  
{  
clrscr();  
cout<<"\a\nPease re-enter a correct option (Y/N) : ";  
clreol();  
cin>>answer;  
}  
cout<<"\n";  
T1.count++;  
T1.serial++;  
} //end of "for loop" for entering information  
clrscr();  
clrscr();  
A1.tasks();  
} //end of function

void dob::enter\_date()  
{  
clreol();  
cout<<"\n\n \*\*\* Date of birth \*\*\*"<<"\n";  
clreol();  
cout<<"\nYear : ";  
clreol();  
clreol();  
cin>>DOB11[T1.temp].year;  
while(DOB11[T1.temp].year>T1.current\_year) //to ensure that the year of birth  
{ //gets entered correctly  
clreol();  
cout<<"\aInvalid input for year !!!";  
cout<<"\nThe date of birth cannot be after the current system date !!! ";  
cout<<"\nPlease enter the year correctly : ";  
clreol();  
cin>>DOB11[T1.temp].year;  
cout<<"\n";  
}  
while(DOB11[T1.temp].year<1000)  
{  
clreol();  
cout<<"Invalid input for year !!! ";  
clreol();  
cout<<"\nThe year of birth must be of 4 digits!!!";  
clreol();  
cout<<"\nPlease enter the year correctly : \a";  
clreol();  
cin>>DOB11[T1.temp].year;  
cout<<"\n";  
}  
while(DOB11[T1.temp].year<1900)  
{  
clreol();  
cout<<"\a\nAre you sure (Y/N) ? ";  
cin>>birth\_answer;  
cout<<"\n";  
while(birth\_answer!='Y'&&birth\_answer!='y'&&birth\_answer!='N'&&birth\_answer!='n')  
{  
clreol();  
cout<<"\a\nInvalid input !!! "<<"\n";  
clreol();  
cout<<"Please enter a correct option (Y/N) : ";  
cout<<"\n";  
clreol();  
cin>>birth\_answer;  
}  
if(birth\_answer=='n'||birth\_answer=='N')  
{  
clreol();  
cout<<"\nPlease enter the year again : ";  
clreol();  
cin>>DOB11[T1.temp].year;  
clreol();  
while(DOB11[T1.temp].year>T1.current\_year)  
{  
clreol();  
cout<<"Invalid input for year !!! ";  
clreol();  
cout<<"\nThe date of birth cannot be after the current system date !!! ";  
clreol();  
cout<<"\nPlease enter the year correctly : \a";  
clreol();  
cin>>DOB11[T1.temp].year;  
cout<<"\n";  
}  
while(DOB11[T1.temp].year<1000)  
{  
clreol();  
cout<<"\aInvalid input for year !!! ";  
clreol();  
cout<<"\nThe year of birth must be of 4 digits!!!\a";  
clreol();  
cout<<"\nPlease enter the year correctly : ";  
clreol();  
cin>>DOB11[T1.temp].year;  
cout<<"\n";  
}  
} //end of inner while  
while(birth\_answer=='y'||birth\_answer=='Y')  
{  
goto jump;  
} //end of inner while  
} //end of outer while  
jump:  
clreol();  
cout<<"Month : ";  
clreol();  
cin>>DOB11[T1.temp].month;  
while(DOB11[T1.temp].month<=0||DOB11[T1.temp].month>12)  
{  
clreol();  
cout<<"\a\nInvalid input for month !!!"<<"\n";  
cout<<"Again enter the month : ";  
clreol();  
clreol();  
cin>>DOB11[T1.temp].month;  
}  
cout<<"Date : ";  
clreol();  
switch(DOB11[T1.temp].month)  
{  
case 1:  
case 3:  
case 5:  
case 7:  
case 8:  
case 10:  
case 12:{  
cin>>DOB11[T1.temp].date; //for months with 31 days  
while(DOB11[T1.temp].date<1||DOB11[T1.temp].date>31)  
{  
clreol();  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : ";  
clreol();  
cin>>DOB11[T1.temp].date;  
cout<<"\n";  
}  
break;  
}  
case 2:{ //for february  
cin>>DOB11[T1.temp].date;  
if(DOB11[T1.temp].year%4==0)  
{  
while(DOB11[T1.temp].date<0||DOB11[T1.temp].date>29) //for leap year  
{  
clreol();  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : ";  
clreol();  
cin>>DOB11[T1.temp].date;  
cout<<"\n";  
}  
}  
else  
{  
while(DOB11[T1.temp].date<0||DOB11[T1.temp].date>28) //for non-leap year  
{  
clreol();  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : ";  
clreol();  
cin>>DOB11[T1.temp].date;  
cout<<"\n";  
}  
}  
break;  
}  
default:{  
cin>>DOB11[T1.temp].date; //for other months  
while(DOB11[T1.temp].date<1||DOB11[T1.temp].date>30)  
{  
clreol();  
cout<<"\aInvalid date !!!"<<"\n";  
cout<<"Again enter the date : ";  
clreol();  
cin>>DOB11[T1.temp].date;  
cout<<"\n";  
}  
break;  
}  
} //end of switch  
clreol();  
} //end of function

void all::show\_patient\_detail()  
{  
clrscr();  
if(T1.d\_index==1)  
{  
T1.d\_index=0;  
cout<<"\n\n";  
goto direct;  
}  
if(T1.current==0)  
{  
cout<<"\n\n\n";  
cout<<"\tDatabase in empty !!!"<<"\n";  
cout<<"\tPlease enter some information first. "<<"\n";  
cout<<"\n\n\tDo you want to enter now (Y/N) : \a";  
cin>>T1.enter\_now;  
while(T1.enter\_now!='Y'&&T1.enter\_now!='y'&&T1.enter\_now!='N'&&T1.enter\_now!='n')  
{  
cout<<"\a\n\tPlease enter a correct option (Y/N) : ";  
cin>>T1.enter\_now;  
}  
if(T1.enter\_now=='y'||T1.enter\_now=='Y')  
{  
A1.enter\_patient\_info();  
}  
else  
{  
A1.tasks();  
}  
}  
cout<<"\n\t\t\*\*\*\*\*\* CURRENT DATEBASE ENTRIES \*\*\*\*\*\*\n\n";  
cout<<"\t\tRegistration no. Name of patient\n\n";  
for(T1.j=1;T1.j<100;T1.j++)  
{  
if(T1.entry\_index[T1.j]==1)  
{  
cout<<"\t\t "<<T1.j<<" "<<PI[T1.j].name<<"\n";  
}  
else  
{  
continue;  
}  
}  
cout<<"\n\n\t\tPress 0 to return to main menu.... ";  
cout<<"\n\n\n\tEnter registration number : ";  
clreol();  
cin>>T1.regis;  
clrscr();  
cout<<"\n\n";  
cin.get(ch);  
if(T1.regis==0)  
{  
A1.tasks();  
}   
T1.show\_count++;  
direct: //label for "goto" jump statement  
if(T1.regis<=0)  
{  
T1.attempt++;  
if(T1.attempt==3)  
{  
cout<<"\n\n\tYou have entered wrong registration number 3 times ."<<"\n";  
cout<<"\tAccess Denied!!! "<<"\n";  
cout<<"\tPlease try again later. "<<"\n";  
cout<<"\tPress any key to exit to main task menu..."<<"\n";  
getch();  
clrscr();  
A1.tasks();  
}  
cout<<"\n\n\tSorry, invalid registration number !!! ";  
cout<<"\n\tRegistration number starts from 1 . ";  
cout<<"\n\n\tPress any key to continue..... \a";  
getch();  
A1.show\_patient\_detail();  
}  
if(T1.regis>0&&T1.regis<T1.serial&&T1.entry\_index[T1.regis]==0)  
{  
cout<<"\n\tSorry, the patient has left the hospital.";  
T1.attempt++;  
if(T1.attempt==3)  
{  
cout<<"\n\n\tYou have entered wrong registration number 3 times ."<<"\n";  
cout<<"\tAccess Denied!!! "<<"\n";  
cout<<"\tPlease try again later. "<<"\n";  
cout<<"\tPress any key to exit to main task menu..."<<"\n";  
getch();  
clrscr();  
A1.tasks();  
}  
cout<<"\n\tPress any key to continue..."<<"\n";  
getch();  
}  
if((T1.regis<0||T1.regis>=T1.serial)&&T1.entry\_index[T1.regis]==-1)  
{  
T1.attempt++;  
clreol();  
if(T1.attempt!=3)  
{  
cout<<"\n\n\tSorry, the registration number is invalid ."<<"\n";  
T1.entry=T1.serial-1;  
if(T1.entry==1)  
{  
cout<<"\tOnly 1 record has been added .\n";  
}  
else  
{  
cout<<"\tOnly "<<T1.entry<<" records have been added .\n";  
}  
}  
if(T1.attempt==3)  
{  
cout<<"\n\n\tYou have entered wrong registration number 3 times ."<<"\n";  
cout<<"\tAccess Denied!!! "<<"\n";  
cout<<"\tPlease try again later. "<<"\n";  
cout<<"\tPress any key to exit to main task menu..."<<"\a\n";  
getch();  
clrscr();  
A1.tasks();  
}  
cout<<"\n\n\n\tPress any key to continue.....";  
getch();  
A1.show\_patient\_detail();  
}  
if(T1.regis>0&&T1.regis<T1.serial&&T1.entry\_index[T1.regis]==1)  
{  
T1.attempt=0;  
clreol();  
cout<<"\n\t \*\*\*INFORMATION FOR PATIENT REGISTRATION NUMBER "<<T1.regis<<"\*\*\*\n\n";  
clreol();  
cout<<"\tName : "<<PI[T1.regis].name<<"\n";  
clreol();  
cout<<"\tSex : ";  
clreol();  
if(PI[T1.regis].sex==1)  
{  
cout<<"Male "<<"\n";  
clreol();  
}  
if(PI[T1.regis].sex==2)  
{  
cout<<"Female "<<"\n";  
clreol();  
}  
cout<<"\tBlood Group : ";  
clreol();  
switch(PI[T1.regis].bld\_group)  
{  
case 1:{  
clreol();  
cout<<"A+\n";  
break;  
}  
case 2:{  
clreol();  
cout<<"A-\n";  
break;  
}  
case 3:{  
clreol();  
cout<<"B+\n";  
break;  
}  
case 4:{  
clreol();  
cout<<"B-\n";  
break;  
}  
case 5:{  
clreol();  
cout<<"AB+\n";  
break;  
}  
case 6:{  
clreol();  
cout<<"AB-\n";  
break;  
}  
case 7:{  
clreol();  
cout<<"O+\n";  
break;  
}  
case 8:{  
clreol();  
cout<<"O-\n";  
break;  
}  
} //end of switch  
clreol();  
cout<<"\tDate of birth : ";  
clreol();  
DOB1.show\_date();  
cout<<"\tMartial Status : ";  
clreol();  
if(PI[T1.regis].martial\_status==1)  
{  
cout<<"Married "<<"\n";  
clreol();  
}  
else  
{  
cout<<"Single "<<"\n";  
clreol();  
}  
clreol();  
cout<<"\n\t \*\*ADDRESS\*\*"<<"\n";  
clreol();  
cout<<"\n\tHouse no. : "<<PI[T1.regis].AD1.house;  
clreol();  
cout<<"\n\tStreet : "<<PI[T1.regis].AD1.street;  
clreol();  
cout<<"\n\tCity : "<<PI[T1.regis].AD1.city;  
clreol();  
cout<<"\n\tDistrict : "<<PI[T1.regis].AD1.dist;  
clreol();  
cout<<"\n\tState : "<<PI[T1.regis].AD1.state;  
clreol();  
cout<<"\n\tCountry : "<<PI[T1.regis].AD1.country;  
clreol();  
}  
T1.d\_index=0;  
clreol();  
clreol();  
cout<<"\n\n\n\n\tWant to view detail of another patient : ";  
clreol();  
cin>>answer1;  
while(answer1!='Y'&&answer1!='y'&&answer1!='N'&&answer1!='n')  
{  
clreol();  
cout<<"\a\tPease re-enter a correct option (Y/N) ? ";  
clreol();  
cin>>answer1;  
}  
clrscr();  
if(answer1=='Y'||answer1=='y')  
{  
A1.show\_patient\_detail();  
}  
else  
{  
A1.tasks();  
}  
} //end of function

void dob::show\_date()  
{  
cout<<DOB11[T1.regis].date;  
T1.rem=DOB11[T1.regis].date%10;  
switch(DOB11[T1.regis].date)  
{  
case 11:  
case 12:  
case 13:  
case 14:  
case 15:  
case 16:  
case 17:  
case 18:  
case 19:  
case 20:{  
cout<<"th ";  
goto over;  
}  
} //end of switch  
switch(T1.rem)  
{  
case 1:{  
cout<<"st ";  
break;  
}  
case 2:{  
cout<<"nd ";  
break;  
}  
case 3:{  
cout<<"rd ";  
break;  
}  
default:{  
cout<<"th ";  
break;  
}  
} //end of switch  
over:  
switch(DOB11[T1.regis].month)  
{  
case 1:{  
cout<<"January , ";  
break;  
}  
case 2:{  
cout<<"February , ";  
break;  
}  
case 3:{  
cout<<"March , ";  
break;  
}  
case 4:{  
cout<<"April , ";  
break;  
}  
case 5:{  
cout<<"May , ";  
break;  
}  
case 6:{  
cout<<"June , ";  
break;  
}  
case 7:{  
cout<<"July , ";  
break;  
}  
case 8:{  
cout<<"August , ";  
break;  
}  
case 9:{  
cout<<"September , ";  
break;  
}  
case 10:{  
cout<<"October , ";  
break;  
}