//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// PROJECT SALARY MANAGMENT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INCLUDED HEADER FILES

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <fstream .h>

#include <process .h>

#include <string .h>

#include <stdlib .h>

#include <stdio .h>

#include <ctype .h>

#include <conio .h>

#include <dos .h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTAINS ALL THE DRAWING FUNCTIONS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class LINES

{

public :

void LINE\_HOR(int, int, int, char) ;

void LINE\_VER(int, int, int, char) ;

void BOX(int,int,int,int,char) ;

void CLEARUP(void) ;

void CLEARDOWN(void) ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTROL ALL THE FUNCTIONS IN THE MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class MENU

{

public :

void MAIN\_MENU(void) ;

private :

void EDIT\_MENU(void) ;

void INTRODUCTION(void) ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTROL ALL THE FUNCTIONS RELATED TO EMPLOYEE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class EMPLOYEE

{

public :

void NEW\_EMPLOYEE(void) ;

void MODIFICATION(void) ;

void DELETION(void) ;

void DISPLAY(void) ;

void LIST(void) ;

void SALARY\_SLIP(void) ;

private :

void ADD\_RECORD(int, char[], char[], char[], int, int, int, char[], char, char, char, float, float) ;

void MODIFY\_RECORD(int, char [], char [], char [], char [], char, char, char, float, float) ;

void DELETE\_RECORD(int) ;

int LASTCODE(void) ;

int CODEFOUND(int) ;

int RECORDNO(int) ;

int FOUND\_CODE(int) ;

void DISPLAY\_RECORD(int) ;

int VALID\_DATE(int, int, int) ;

int code, dd, mm, yy ;

char name[26], address[31], phone[10], desig[16] ;

char grade, house, convense ;

float loan, basic ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CONTROL ALL THE FUNCTIONS IN THE MAIN MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void MENU :: MAIN\_MENU(void)

{

char ch ;

LINES L ;

L.CLEARUP() ;

while (1)

{ int textcolor[15];

int textbackground[1];

clrscr() ;

L.BOX(28,7,51,9,218) ;

L.BOX(10,5,71,21,218) ;

L.BOX(11,6,70,20,219) ;

gotoxy(31,8) ;

cout < <" INDIAN PVT. LTD." ;

gotoxy(30,11) ;

cout <<"1: NEW EMPLOYEE" ;

gotoxy(30,12) ;

cout <<"2: DISPLAY EMPLOYEE" ;

gotoxy(30,13) ;

cout <<"3: LIST OF EMPLOYEES" ;

gotoxy(30,14) ;

cout <<"4: SALARY SLIP" ;

gotoxy(30,15) ;

cout <<"5: EDIT" ;

gotoxy(30,16) ;

cout <<"0: QUIT" ;

gotoxy(30,18) ;

cout <<"ENTER YOUR CHOICE :" ;

ch = getch() ;

if (ch == 27 || ch == '0')

break ;

else

if (ch == '1')

{

EMPLOYEE E ;

E.NEW\_EMPLOYEE() ;

}

else

if (ch == '2')

{

EMPLOYEE E ;

E.DISPLAY() ;

}

else

if (ch == '3')

{

EMPLOYEE E ;

E.LIST() ;

}

else

if (ch == '4')

{

EMPLOYEE E ;

E.SALARY\_SLIP() ;

}

else

if (ch == '5')

EDIT\_MENU() ;

}

L.CLEARUP() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CONTROL ALL THE FUNCTIONS IN THE EDIT MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void MENU :: EDIT\_MENU(void)

{

char ch ;

LINES L ;

L.CLEARDOWN() ;

while (1)

{

clrscr() ;

L.BOX(28,8,49,10,218) ;

L.BOX(10,5,71,21,218) ;

L.BOX(11,6,70,20,219) ;

gotoxy(31,9) ;

cout <<"E D I T M E N U" ;

gotoxy(30,13) ;

cout <<"1: DELETE RECORD" ;

gotoxy(30,14) ;

cout <<"2: MODIFY RECORD" ;

gotoxy(30,15) ;

cout <<"0: EXIT" ;

gotoxy(30,17) ;

cout <<"ENTER YOUR CHOICE :" ;

ch = getch() ;

if (ch == 27 || ch == '0')

break ;

else

if (ch == '1')

{

EMPLOYEE E ;

E.DELETION() ;

}

else

if (ch == '2')

{

EMPLOYEE E ;

E.MODIFICATION() ;

}

}

L.CLEARDOWN() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE HORRIZONTAL LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINES :: LINE\_HOR(int column1, int column2, int row, char c)

{

for ( column1; column1<=column2; column1++ )

{

gotoxy(column1,row) ;

cout <<c ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE VERTICAL LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINES :: LINE\_VER(int row1, int row2, int column, char c)

{

for ( row1; row1<=row2; row1++ )

{

gotoxy(column,row1) ;

cout <<c ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE BOX

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINES :: BOX(int column1, int row1, int column2, int row2, char c)

{

char ch=218 ;

char c1, c2, c3, c4 ;

char l1=196, l2=179 ;

if (c == ch)

{

c1=218 ;

c2=191 ;

c3=192 ;

c4=217 ;

l1 = 196 ;

l2 = 179 ;

}

else

{

c1=c ;

c2=c ;

c3=c ;

c4=c ;

l1 = c ;

l2 = c ;

}

gotoxy(column1,row1) ;

cout <<c1 ;

gotoxy(column2,row1) ;

cout <<c2 ;

gotoxy(column1,row2) ;

cout <<c3 ;

gotoxy(column2,row2) ;

cout <<c4 ;

column1++ ;

column2-- ;

LINE\_HOR(column1,column2,row1,l1) ;

LINE\_HOR(column1,column2,row2,l1) ;

column1-- ;

column2++ ;

row1++ ;

row2-- ;

LINE\_VER(row1,row2,column1,l2) ;

LINE\_VER(row1,row2,column2,l2) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CLEAR THE SCREEN LINE BY LINE UPWARD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINES :: CLEARUP(void)

{

for (int i=25; i>=1; i--)

{

//delay(20) ;

gotoxy(1,i) ; clreol() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CLEAR THE SCREEN LINE BY LINE DOWNWORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINES :: CLEARDOWN(void)

{

for (int i=1; i< =25; i++)

{

//delay(20) ;

gotoxy(1,i) ; clreol() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ADDS THE GIVEN DATA IN THE EMPLOYEE'S FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: ADD\_RECORD(int ecode, char ename[26], char eaddress[31], char ephone[10], int d, int m, int y, char edesig[16], char egrade, char ehouse, char econv, float eloan, float ebasic)

{

fstream file ;

file.open("EMPLOYEE.DAT", ios::app) ;

code = ecode ;

strcpy(name,ename) ;

strcpy(address,eaddress) ;

strcpy(phone,ephone) ;

dd = d ;

mm = m ;

yy = y ;

strcpy(desig,edesig) ;

grade = egrade ;

house = ehouse ;

convense = econv ;

loan = eloan ;

basic = ebasic ;

file.write((char \*) this, sizeof(EMPLOYEE)) ;

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION MODIFY THE GIVEN DATA IN THE

// EMPLOYEE'S FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: MODIFY\_RECORD(int ecode, char ename[26], char eaddress[31], char ephone[10], char edesig[16], char egrade, char ehouse, char econv, float eloan, float ebasic)

{

int recno ;

recno = RECORDNO(ecode) ;

fstream file ;

file.open("EMPLOYEE.DAT", ios::out | ios::ate) ;

strcpy(name,ename) ;

strcpy(address,eaddress) ;

strcpy(phone,ephone) ;

strcpy(desig,edesig) ;

grade = egrade ;

house = ehouse ;

convense = econv ;

loan = eloan ;

basic = ebasic ;

int location ;

location = (recno-1) \* sizeof(EMPLOYEE) ;

file.seekp(location) ;

file.write((char \*) this, sizeof(EMPLOYEE)) ;

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DELETE THE RECORD IN THE EMPLOYEE FILE

// FOR THE GIVEN EMPLOYEE CODE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: DELETE\_RECORD(int ecode)

{

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

fstream temp ;

temp.open("temp.dat", ios::out) ;

file.seekg(0,ios::beg) ;

while (!file.eof())

{

file.read((char \*) this, sizeof(EMPLOYEE)) ;

if (file.eof())

break ;

if (code != ecode)

temp.write((char \*) this, sizeof(EMPLOYEE)) ;

}

file.close() ;

temp.close() ;

file.open("EMPLOYEE.DAT", ios::out) ;

temp.open("temp.dat", ios::in) ;

temp.seekg(0,ios::beg) ;

while (!temp.eof())

{

temp.read((char \*) this, sizeof(EMPLOYEE)) ;

if ( temp.eof() )

break ;

file.write((char \*) this, sizeof(EMPLOYEE)) ;

}

file.close() ;

temp.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS THE LAST EMPLOYEE'S CODE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int EMPLOYEE :: LASTCODE(void)

{

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int count=0 ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

count = code ;

file.close() ;

return count ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS 0 IF THE GIVEN CODE NOT FOUND

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int EMPLOYEE :: FOUND\_CODE(int ecode)

{

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int found=0 ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

{

if (code == ecode)

{

found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS RECORD NO. OF THE GIVEN CODE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int EMPLOYEE :: RECORDNO(int ecode)

{

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int recno=0 ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

{

recno++ ;

if (code == ecode)

break ;

}

file.close() ;

return recno ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAYS THE LIST OF THE EMPLOYEES

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: LIST(void)

{

clrscr() ;

int row = 6 , found=0, flag=0 ;

char ch ;

gotoxy(31,2) ;

cout <<"LIST OF EMPLOYEES" ;

gotoxy(30,3) ;

cout <<"~~~~~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ;

cout <<"CODE NAME PHONE DOJ DESIGNATION GRADE SALARY" ;

gotoxy(1,5) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

{

flag = 0 ;

//delay(20) ;

found = 1 ;

gotoxy(2,row) ;

cout <<code ;

gotoxy(6,row) ;

cout <<name ;

gotoxy(31,row) ;

cout <<phone ;

gotoxy(40,row) ;

cout <<dd <<"/" <<mm <<"/" <<yy ;

gotoxy(52,row) ;

cout <<desig ;

gotoxy(69,row) ;

cout <<grade ;

if (grade != 'E')

{

gotoxy(74,row) ;

cout <<basic ;

}

else

{

gotoxy(76,row) ;

cout <<"-" ;

}

if ( row == 23 )

{

flag = 1 ;

row = 6 ;

gotoxy(1,25) ;

cout <<"Press any key to continue or Press <ESC> to exit" ;

ch = getch() ;

if (ch == 27)

break ;

clrscr() ;

gotoxy(31,2) ;

cout < <"LIST OF EMPLOYEES" ;

gotoxy(30,3) ;

cout <<"~~~~~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ;

cout <<"CODE NAME PHONE DOJ DESIGNATION GRADE SALARY" ;

gotoxy(1,5) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

}

else

row++ ;

}

if (!found)

{

gotoxy(5,10) ;

cout <<"**\7**Records not found" ;

}

if (!flag)

{

gotoxy(1,25) ;

cout <<"Press any key to continue..." ;

getche() ;

}

file.close () ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAYS THE RECORD OF THE EMPLOYEES

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: DISPLAY\_RECORD(int ecode)

{

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

{

if (code == ecode)

{

gotoxy(5,5) ;

cout <<"Employee Code # " <<code ;

gotoxy(5,6) ;

cout <<"~~~~~~~~~~~~~" ;

gotoxy(5,7) ;

cout <<"Name : " <<name ;

gotoxy(5,8) ;

cout <<"Address : " <<address ;

gotoxy(5,9) ;

cout <<"Phone no. : " <<phone ;

gotoxy(5,11) ;

cout <<"JOINING DATE" ;

gotoxy(5,12) ;

cout <<"~~~~~~~~~~~~" ;

gotoxy(5,13) ;

cout <<"Day : " <<dd ;

gotoxy(5,14) ;

cout <<"Month : " <<mm ;

gotoxy(5,15) ;

cout <<"Year : " <<yy ;

gotoxy(5,17) ;

cout <<"Designation : " <<desig ;

gotoxy(5,18) ;

cout <<"Grade : " <<grade ;

if (grade != 'E')

{

gotoxy(5,19) ;

cout <<"House (y/n) : " <<house ;

gotoxy(5,20) ;

cout <<"Convense (y/n) : " <<convense ;

gotoxy(5,22) ;

cout <<"Basic Salary : " <<basic ;

}

gotoxy(5,21) ;

cout <<"Loan : " <<loan ;

}

}

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION GIVE DATA TO ADD IN THE FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: NEW\_EMPLOYEE(void)

{

clrscr() ;

char ch, egrade, ehouse='N', econv='N' ;

char ename[26], eaddress[31], ephone[10], edesig[16], t1[10] ;

float t2=0.0, eloan=0.0, ebasic=0.0 ;

int d, m, y, ecode, valid ;

gotoxy(72,2) ;

cout <<"&lt;0>=EXIT" ;

gotoxy(28,3) ;

cout < <"ADDITION OF NEW EMPLOYEE" ;

gotoxy(5,5) ;

cout <<"Employee Code # " ;

gotoxy(5,6) ;

cout <<"~~~~~~~~~~~~~" ;

gotoxy(5,7) ;

cout <<"Name : " ;

gotoxy(5,8) ;

cout <<"Address : " ;

gotoxy(5,9) ;

cout <<"Phone no. : " ;

gotoxy(5,11) ;

cout <<"JOINING DATE" ;

gotoxy(5,12) ;

cout <<"~~~~~~~~~~~~" ;

gotoxy(5,13) ;

cout <<"Day : " ;

gotoxy(5,14) ;

cout <<"Month : " ;

gotoxy(5,15) ;

cout <<"Year : " ;

gotoxy(5,17) ;

cout <<"Designation : " ;

gotoxy(5,18) ;

cout <<"Grade : " ;

gotoxy(5,21) ;

cout <<"Loan : " ;

ecode = LASTCODE() + 1 ;

if (ecode == 1)

{

ADD\_RECORD(ecode, "null", "null", "null", 0, 0, 0, "null", 'n', 'n', 'n', 0.0, 0.0) ;

DELETE\_RECORD(ecode) ;

}

gotoxy(21,5) ;

cout <<ecode ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the name of the Employee" ;

gotoxy(20,7) ; clreol() ;

gets(ename) ;

strupr(ename) ;

if (ename[0] == '0')

return ;

if (strlen(ename) < 1 || strlen(ename) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly (Range: 1..25)" ;

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Address of the Employee" ;

gotoxy(20,8) ; clreol() ;

gets(eaddress) ;

strupr(eaddress) ;

if (eaddress[0] == '0')

return ;

if (strlen(eaddress) < 1 || strlen(eaddress) > 30)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly (Range: 1..30)" ;

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Phone no. of the Employee or Press <ENTER> for none" ;

gotoxy(20,9) ; clreol() ;

gets(ephone) ;

if (ephone[0] == '0')

return ;

if ((strlen(ephone) < 7 && strlen(ephone) > 0) || (strlen(ephone) > 9))

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly" ;

getch() ;

}

} while (!valid) ;

if (strlen(ephone) == 0)

strcpy(ephone,"-") ;

char tday[3], tmonth[3], tyear[5] ;

int td ;

do

{

valid = 1 ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ENTER DAY OF JOINING" ;

gotoxy(13,13) ; clreol() ;

gets(tday) ;

td = atoi(tday) ;

d = td ;

if (tday[0] == '0')

return ;

} while (d == 0) ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ENTER MONTH OF JOINING" ;

gotoxy(13,14) ; clreol() ;

gets(tmonth) ;

td = atoi(tmonth) ;

m = td ;

if (tmonth[0] == '0')

return ;

} while (m == 0) ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ENTER YEAR OF JOINING" ;

gotoxy(13,15) ; clreol() ;

gets(tyear) ;

td = atoi(tyear) ;

y = td ;

if (tyear[0] == '0')

return ;

} while (y == 0) ;

if (d>31 || d&lt;1)

valid = 0 ;

else

if (((y%4)!=0 && m==2 && d>28) || ((y%4)==0 && m==2 && d>29))

valid = 0 ;

else

if ((m==4 || m==6 || m==9 || m==11) && d>30)

valid = 0 ;

else

if (y&lt;1990 || y>2020)

valid = 0 ;

if (!valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly" ;

getch() ;

gotoxy(13,14) ; clreol() ;

gotoxy(13,15) ; clreol() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Designation of the Employee" ;

gotoxy(20,17) ; clreol() ;

gets(edesig) ;

strupr(edesig) ;

if (edesig[0] == '0')

return ;

if (strlen(edesig) < 1 || strlen(edesig) > 15)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly (Range: 1..15)" ;

getch() ;

}

} while (!valid) ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"Enter Grade of the Employee (A,B,C,D,E)" ;

gotoxy(20,18) ; clreol() ;

egrade = getche() ;

egrade = toupper(egrade) ;

if (egrade == '0')

return ;

} while (egrade < 'A' || egrade > 'E') ;

if (egrade != 'E')

{

gotoxy(5,19) ;

cout < <"House (y/n) : " ;

gotoxy(5,20) ;

cout <<"Convense (y/n) : " ;

gotoxy(5,22) ;

cout <<"Basic Salary : " ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ENTER IF HOUSE ALLOWANCE IS ALLOTED TO EMPLOYEE OR NOT" ;

gotoxy(22,19) ; clreol() ;

ehouse = getche() ;

ehouse = toupper(ehouse) ;

if (ehouse == '0')

return ;

} while (ehouse != 'Y' && ehouse != 'N') ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ENTER IF CONVENCE ALLOWANCE IS ALLOTED TO EMPLOYEE OR NOT" ;

gotoxy(22,20) ; clreol() ;

econv = getche() ;

econv = toupper(econv) ;

if (econv == '0')

return ;

} while (econv != 'Y' && econv != 'N') ;

}

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"ENTER LOAN AMOUNT IF ISSUED" ;

gotoxy(22,21) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

eloan = t2 ;

if (eloan > 50000)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**SHOULD NOT GREATER THAN 50000" ;

getch() ;

}

} while (!valid) ;

if (egrade != 'E')

{

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"ENTER BASIC SALARY OF THE EMPLOYEE" ;

gotoxy(22,22) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

ebasic = t2 ;

if (t1[0] == '0')

return ;

if (ebasic > 50000)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**SHOULD NOT GREATER THAN 50000" ;

getch() ;

}

} while (!valid) ;

}

gotoxy(5,25) ; clreol() ;

do

{

gotoxy(5,24) ; clreol() ;

cout <<"Do you want to save (y/n) " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

ADD\_RECORD(ecode, ename, eaddress, ephone, d, m, y, edesig, egrade, ehouse, econv, eloan, ebasic) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION GIVE CODE FOR THE DISPLAY OF THE RECORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: DISPLAY(void)

{

clrscr() ;

char t1[10] ;

int t2, ecode ;

gotoxy(72,2) ;

cout <<"&lt;0>=EXIT" ;

gotoxy(5,5) ;

cout < <"Enter code of the Employee " ;

gets(t1) ;

t2 = atoi(t1) ;

ecode = t2 ;

if (ecode == 0)

return ;

clrscr() ;

if (!FOUND\_CODE(ecode))

{

gotoxy(5,5) ;

cout <<"**\7**Record not found" ;

getch() ;

return ;

}

DISPLAY\_RECORD(ecode) ;

gotoxy(5,25) ;

cout <<"Press any key to continue..." ;

getch() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION GIVE DATA FOR THE MODIFICATION OF THE

// EMPLOYEE RECORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: MODIFICATION(void)

{

clrscr() ;

char ch, egrade, ehouse='N', econv='N' ;

char ename[26], eaddress[31], ephone[10], edesig[16], t1[10] ;

float t2=0.0, eloan=0.0, ebasic=0.0 ;

int ecode, valid ;

gotoxy(72,2) ;

cout <<"&lt;0>=EXIT" ;

gotoxy(5,5) ;

cout < <"Enter code of the Employee " ;

gets(t1) ;

t2 = atoi(t1) ;

ecode = t2 ;

if (ecode == 0)

return ;

clrscr() ;

if (!FOUND\_CODE(ecode))

{

gotoxy(5,5) ;

cout <<"**\7**Record not found" ;

getch() ;

return ;

}

gotoxy(72,2) ;

cout <<"&lt;0>=EXIT" ;

gotoxy(22,3) ;

cout < <"MODIFICATION OF THE EMPLOYEE RECORD" ;

DISPLAY\_RECORD(ecode) ;

do

{

gotoxy(5,24) ; clreol() ;

cout <<"Do you want to modify this record (y/n) " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

clrscr() ;

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

{

if (code == ecode)

break ;

}

file.close() ;

gotoxy(5,5) ;

cout <<"Employee Code # " <<ecode ;

gotoxy(5,6) ;

cout <<"~~~~~~~~~~~~~" ;

gotoxy(40,5) ;

cout <<"JOINING DATE : " ;

gotoxy(40,6) ;

cout <<"~~~~~~~~~~~~~~" ;

gotoxy(55,5) ;

cout <<dd <<"/" <<mm <<"/" <<yy ;

gotoxy(5,7) ;

cout <<"Name : " ;

gotoxy(5,8) ;

cout <<"Address : " ;

gotoxy(5,9) ;

cout <<"Phone no. : " ;

gotoxy(5,10) ;

cout <<"Designation : " ;

gotoxy(5,11) ;

cout <<"Grade : " ;

gotoxy(5,14) ;

cout <<"Loan : " ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the name of the Employee or <ENTER> FOR NO CHANGE" ;

gotoxy(20,7) ; clreol() ;

gets(ename) ;

strupr(ename) ;

if (ename[0] == '0')

return ;

if (strlen(ename) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly (Range: 1..25)" ;

getch() ;

}

} while (!valid) ;

if (strlen(ename) == 0)

{

strcpy(ename,name) ;

gotoxy(20,7) ;

cout <<ename ;

}

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Address of the Employee or <ENTER> FOR NO CHANGE" ;

gotoxy(20,8) ; clreol() ;

gets(eaddress) ;

strupr(eaddress) ;

if (eaddress[0] == '0')

return ;

if (strlen(eaddress) > 30)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly (Range: 1..30)" ;

getch() ;

}

} while (!valid) ;

if (strlen(eaddress) == 0)

{

strcpy(eaddress,address) ;

gotoxy(20,8) ;

cout <<eaddress ;

}

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Phone no. of the Employee or or <ENTER> FOR NO CHANGE" ;

gotoxy(20,9) ; clreol() ;

gets(ephone) ;

if (ephone[0] == '0')

return ;

if ((strlen(ephone) < 7 && strlen(ephone) > 0) || (strlen(ephone) > 9))

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly" ;

getch() ;

}

} while (!valid) ;

if (strlen(ephone) == 0)

{

strcpy(ephone,phone) ;

gotoxy(20,9) ;

cout <<ephone ;

}

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Designation of the Employee or <ENTER> FOR NO CHANGE" ;

gotoxy(20,10) ; clreol() ;

gets(edesig) ;

strupr(edesig) ;

if (edesig[0] == '0')

return ;

if (strlen(edesig) > 15)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**Enter correctly (Range: 1..15)" ;

getch() ;

}

} while (!valid) ;

if (strlen(edesig) == 0)

{

strcpy(edesig,desig) ;

gotoxy(20,10) ;

cout <<edesig ;

}

do

{

gotoxy(5,25) ; clreol() ;

cout <<"Enter Grade of the Employee (A,B,C,D,E) or <ENTER> FOR NO CHANGE" ;

gotoxy(20,11) ; clreol() ;

egrade = getche() ;

egrade = toupper(egrade) ;

if (egrade == '0')

return ;

if (egrade == 13)

{

egrade = grade ;

gotoxy(20,11) ;

cout < <grade ;

}

} while (egrade < 'A' || egrade > 'E') ;

if (egrade != 'E')

{

gotoxy(5,12) ;

cout < <"House (y/n) : " ;

gotoxy(5,13) ;

cout <<"Convense (y/n) : " ;

gotoxy(5,15) ;

cout <<"Basic Salary : " ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ALLOTED HOUSE ALLOWANCE ? or <ENTER> FOR NO CHANGE" ;

gotoxy(22,12) ; clreol() ;

ehouse = getche() ;

ehouse = toupper(ehouse) ;

if (ehouse == '0')

return ;

if (ehouse == 13)

{

ehouse = house ;

gotoxy(22,12) ;

cout < <ehouse ;

}

} while (ehouse != 'Y' && ehouse != 'N') ;

do

{

gotoxy(5,25) ; clreol() ;

cout <<"ALLOTED CONVENCE ALLOWANCE or <ENTER> FOR NO CHANGE" ;

gotoxy(22,13) ; clreol() ;

econv = getche() ;

econv = toupper(econv) ;

if (econv == '0')

return ;

if (econv == 13)

{

econv = convense ;

gotoxy(22,13) ;

cout < <econv ;

}

} while (econv != 'Y' && econv != 'N') ;

}

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"ENTER LOAN AMOUNT or <ENTER> FOR NO CHANGE" ;

gotoxy(22,14) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

eloan = t2 ;

if (eloan > 50000)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**SHOULD NOT GREATER THAN 50000" ;

getch() ;

}

} while (!valid) ;

if (strlen(t1) == 0)

{

eloan = loan ;

gotoxy(22,14) ;

cout <<eloan ;

}

if (egrade != 'E')

{

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"ENTER BASIC SALARY or <ENTER> FOR NO CHANGE" ;

gotoxy(22,15) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

ebasic = t2 ;

if (t1[0] == '0')

return ;

if (ebasic > 50000)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout < <"**\7**SHOULD NOT GREATER THAN 50000" ;

getch() ;

}

} while (!valid) ;

if (strlen(t1) == 0)

{

ebasic = basic ;

gotoxy(22,15) ;

cout <<ebasic ;

}

}

gotoxy(5,25) ; clreol() ;

do

{

gotoxy(5,18) ; clreol() ;

cout <<"Do you want to save (y/n) " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

MODIFY\_RECORD(ecode,ename,eaddress,ephone,edesig,egrade,ehouse,econv,eloan,ebasic) ;

gotoxy(5,23) ;

cout <<"**\7**Record Modified" ;

gotoxy(5,25) ;

cout <<"Press any key to continue..." ;

getch() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION GIVE CODE NO. FOR THE DELETION OF THE

// EMPLOYEE RECORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: DELETION(void)

{

clrscr() ;

char t1[10], ch ;

int t2, ecode ;

gotoxy(72,2) ;

cout <<"&lt;0>=EXIT" ;

gotoxy(5,5) ;

cout < <"Enter code of the Employee " ;

gets(t1) ;

t2 = atoi(t1) ;

ecode = t2 ;

if (ecode == 0)

return ;

clrscr() ;

if (!FOUND\_CODE(ecode))

{

gotoxy(5,5) ;

cout <<"**\7**Record not found" ;

getch() ;

return ;

}

gotoxy(72,2) ;

cout <<"&lt;0>=EXIT" ;

gotoxy(24,3) ;

cout < <"DELETION OF THE EMPLOYEE RECORD" ;

DISPLAY\_RECORD(ecode) ;

do

{

gotoxy(5,24) ; clreol() ;

cout <<"Do you want to delete this record (y/n) " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

DELETE\_RECORD(ecode) ;

LINES L ;

L.CLEARDOWN() ;

gotoxy(5,23) ;

cout <<"**\7**Record Deleted" ;

gotoxy(5,25) ;

cout <<"Press any key to continue..." ;

getch() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN 0 IF THE GIVEN DATE IS INVALID

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int EMPLOYEE :: VALID\_DATE(int d1, int m1, int y1)

{

int valid=1 ;

if (d1>31 || d1&lt;1)

valid = 0 ;

else

if (((y1%4)!=0 && m1==2 && d1>28) || ((y1%4)==0 && m1==2 && d1>29))

valid = 0 ;

else

if ((m1==4 || m1==6 || m1==9 || m1==11) && d1>30)

valid = 0 ;

return valid ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION PRINTS THE SALARY SLIP FOR THE EMPLOYEE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EMPLOYEE :: SALARY\_SLIP(void)

{

clrscr() ;

char t1[10] ;

int t2, ecode, valid ;

gotoxy(72,2) ;

cout < <"&lt;0>=EXIT" ;

gotoxy(5,5) ;

cout < <"Enter code of the Employee " ;

gets(t1) ;

t2 = atoi(t1) ;

ecode = t2 ;

if (ecode == 0)

return ;

clrscr() ;

if (!FOUND\_CODE(ecode))

{

gotoxy(5,5) ;

cout <<"**\7**Record not found" ;

getch() ;

return ;

}

fstream file ;

file.open("EMPLOYEE.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

while (file.read((char \*) this, sizeof(EMPLOYEE)))

{

if (code == ecode)

break ;

}

file.close() ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

char \*mon[12]={"January","February","March","April","May","June","July","August","September","November","December"} ;

LINES L ;

L.BOX(2,1,79,25,219) ;

gotoxy(31,2) ;

cout <<"SAM'S PVT. LTD." ;

L.LINE\_HOR(3,78,3,196) ;

gotoxy(34,4) ;

cout <<"SALARY SLIP" ;

gotoxy(60,4) ;

cout <<"Date: " <<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(34,5) ;

cout <<mon[m1-1] <<", " <<y1 ;

L.LINE\_HOR(3,78,6,196) ;

gotoxy(6,7) ;

cout <<"Employee Name : " <<name ;

gotoxy(6,8) ;

cout <<"Designation : " <<desig ;

gotoxy(67,8) ;

cout <<"Grade : " <<grade ;

L.BOX(6,9,75,22,218) ;

L.LINE\_HOR(10,71,20,196) ;

int days, hours ;

if (grade == 'E')

{

do

{

valid = 1 ;

gotoxy(10,21) ;

cout <<"ENTER NO. OF DAYS WORKED IN THE MONTH " ;

gotoxy(10,11) ;

cout <<"No. of Days : " ;

gets(t1) ;

t2 = atof(t1) ;

days = t2 ;

if (!VALID\_DATE(days,m1,y1))

{

valid = 0 ;

gotoxy(10,21) ;

cout <<"**\7**ENTER CORRECTLY " ;

getch() ;

gotoxy(10,11) ;

cout <<" " ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(10,21) ;

cout <<"ENTER NO. OF HOURS WORKED OVER TIME " ;

gotoxy(10,13) ;

cout <<"No. of hours : " ;

gets(t1) ;

t2 = atof(t1) ;

hours = t2 ;

if (hours > 8 || hours < 0)

{

valid = 0 ;

gotoxy(10,21) ;

cout <<"**\7**ENTER CORRECTLY " ;

getch() ;

gotoxy(10,13) ;

cout <<" " ;

}

} while (!valid) ;

gotoxy(10,21) ;

cout <<" " ;

gotoxy(10,11) ;

cout <<" " ;

gotoxy(10,13) ;

cout <<" " ;

}

gotoxy(10,10) ;

cout <<"Basic Salary : Rs." ;

gotoxy(10,12) ;

cout <<"ALLOWANCE" ;

if (grade != 'E')

{

gotoxy(12,13) ;

cout <<"HRA : Rs." ;

gotoxy(12,14) ;

cout <<"CA : Rs." ;

gotoxy(12,15) ;

cout <<"DA : Rs." ;

}

else

{

gotoxy(12,13) ;

cout <<"OT : Rs." ;

}

gotoxy(10,17) ;

cout <<"DEDUCTIONS" ;

gotoxy(12,18) ;

cout <<"LD : Rs." ;

if (grade != 'E')

{

gotoxy(12,19) ;

cout <<"PF : Rs." ;

}

gotoxy(10,21) ;

cout <<"NET SALARY : Rs." ;

gotoxy(6,24) ;

cout <<"CASHIER" ;

gotoxy(68,24) ;

cout <<"EMPLOYEE" ;

float HRA=0.0, CA=0.0, DA=0.0, PF=0.0, LD=0.0, OT=0.0, allowance, deduction, netsalary ;

if (grade != 'E')

{

if (house == 'Y')

HRA = (5\*basic)/100 ;

if (convense == 'Y')

CA = (2\*basic)/100 ;

DA = (5\*basic)/100 ;

PF = (2\*basic)/100 ;

LD = (15\*loan)/100 ;

allowance = HRA+CA+DA ;

deduction = PF+LD ;

}

else

{

basic = days \* 30 ;

LD = (15\*loan)/100 ;

OT = hours \* 10 ;

allowance = OT ;

deduction = LD ;

}

netsalary = (basic+allowance)-deduction ;

gotoxy(36,10) ;

cout <<basic ;

if (grade != 'E')

{

gotoxy(22,13) ;

cout <<HRA ;

gotoxy(22,14) ;

cout <<CA ;

gotoxy(22,15) ;

cout <<DA ;

gotoxy(22,19) ;

cout <<PF ;

}

else

{

gotoxy(22,13) ;

cout <<OT ;

}

gotoxy(22,18) ;

cout <<LD ;

gotoxy(33,15) ;

cout <<"Rs." <<allowance ;

gotoxy(33,19) ;

cout <<"Rs." <<deduction ;

gotoxy(36,21) ;

cout <<netsalary ;

gotoxy(2,1) ;

getch() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MAIN FUNCTION CALLING MAIN MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main()

{

clrscr();

gotoxy(27,10);

cout<<"JASMEET SINGH";

gotoxy(27,11);

cout<<"Class COMPUTER .";

gotoxy(27,12);

cout<<"Presents";

int i,j,k;

char ch[100];

gotoxy(24,18);

i=18;

cprintf(" PROJECT'2008...");

textcolor(i+1);

gotoxy(24,19);

cprintf("in C++");

textcolor(i+5);

gotoxy(24,20);

cprintf("press enter to continue");

textcolor(i+8);

getch();

clrscr();

gotoxy(30,12);

cprintf("loading ");

j=3;

textcolor(j);

for(i=0;i&lt;8;i++)

{

cprintf(".");

}

struct intro

{

char pro[20];

char pronam[20];

char dir[20];

char dirnam[20];

char gra[20];

char graname[20];

};

MENU menu ;

menu.MAIN\_MENU() ;

}