```
#include<iostream.h>
#include<process.h>
#include<string.h>
#include<stdlib.h>
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<dos.h>
#include<fstream.h>
#include<graphics.h>
#include<math.h>
class MENU
{
     public:
          void mainmenu(void);
          void introduc(void);
          void bkgrnd(int);
          int bkgrnd(void);
          int check(int,int);
     private:
          void editmenu(void);
          void editbook(void);
          void editmem(void);
};
class BOOK
{
     public:
          void list(void);
          char *bookname(int);
     protected:
          void addnewbook(int,char tname[33],char tauthor[26],float,int,int);
          void updatecopies(int,int,int);
          void modify(int,char[],char[],float);
          void deletion(void);
          int bookfound(int);
          int booknamefound(char[]);
          int recordno(int);
          int available(int);
          char *authorname(int);
          float bookprice(int);
          int noofcopies(int);
          int bookcodeof(char[]);
          void display(int);
          int reccount(void);
          void deleterec(int);
     private:
          int bookcode,copies;
          char name[33],author[26];
          float price;
          int avail;
};
class MEMBER
{
     public:
          void list(void);
     protected:
          void addmem(int,int,char[],char[],char[],int,int,int);
          void modify(int,char[],char[],char[]);
```

```
          void deletion(void);
          int memfound(int);
          void updatebook(int,int,int,int,int);
          char *memname(int);
          char *memphone(int);
          char *memaddress(int);
          int recordno(int);
          int lastcode(void);
          int issued(int);
          int fine(int);
          void display(int);
          void deleterec(int);
     private:
          int memcode,bookcode;
          char name[26],phone[10],address[33];
          int dd,mm,yy;
};
class WORKING:public BOOK,public MEMBER
{
     public:
          void issuebook(void);
          void returnbook(void);
          void addbook(void);
          void addmember(void);
          void modifybook(void);
          void modifymem(void);
          void deletebook(void);
          void deletemem(void);
};
class DATE
{
     public:
          void extenddate(int,int,int,int);
          int diff(int,int,int,int,int,int);
          int day,mon,year;
};
void DATE::extenddate(int d1,int m1,int y1,int days)
{
     static int month[]={31,29,31,30,31,30,31,31,30,31,30,31};
     for(int i=1;i<=days;i++)
     {
          d1++;
          if((d1>month[m1-1]) || (y1%4!=0 && m1==2 && d1>28))
          {
               d1=1;
               m1++;
          }
          if(m1>12)
          {
               m1=1;
               y1++;
          }
     }
     day=d1;
     mon=m1;
     year=y1;
}
int DATE::diff(int d1,int m1,int y1,int d2,int m2,int y2)
```

```
{
     int days=0;
     if((y2<y1) || (y2==y1 && m2<m1) || (y2==y1 && m2==m1 && d2<d1))
     return days;
     static int month[]={31,29,31,30,31,30,31,31,30,31,30,31};
     while(d1!=d2 || m1!=m2 || y1!=y2)
     {
          days++;
          d1++;
          if((d1>month[m1-1]) || (y1%4!=0 && m1==2 && d1>28))
          {
               d1=1;
               m1++;
          };
          if(m1>12)
          {
               m1=1;
               y1++;
          }
     }
     return days;
}
void MENU::mainmenu(void)
{

     char ch;
     int p;
     MENU::bkgrnd(0);
     while(1)
     {
           // cleardevice();


          gotoxy(29,6);
          cout<<"BOOK LIBRARY";
          gotoxy(29,7);
          cout<<"-------------------------------";
          gotoxy(30,10);cout<<"1.Introduction";
          gotoxy(30,11);cout<<"2.Add New Book(s)";
          gotoxy(30,12);cout<<"3.Add New Member(s)";
          gotoxy(30,13);cout<<"4.Issue Book(s)";
          gotoxy(30,14);cout<<"5.Return Book(s)";
          gotoxy(30,15);cout<<"6.List of Book(s)";
          gotoxy(30,16);cout<<"7.List of Member(s)";
          gotoxy(30,17);cout<<"8.EDIT";
          gotoxy(30,18);cout<<"0.QUIT";
          gotoxy(30,20);cout<<"Enter Your Choice:";
          ch=getch();
          closegraph();
          if(ch==27)
          break;
          if(ch=='1')
          introduc();
          else
          if(ch=='2')
          {
               WORKING W;
               W.addbook();
```

```
          }
          else
          if(ch=='3')
          {
               WORKING W;
               W.addmember();
          }
          else
          if(ch=='4')
          {
               WORKING W;
               W.issuebook();
          }
          else
          if(ch=='5')
          {
               WORKING W;
               W.returnbook();
          }
          else
          if(ch=='6')
          {
               BOOK B;
               B.list();
          }
          else
          if(ch=='7')
          {
               MEMBER M;
               M.list();
          }
          else
          if(ch=='8')
          editmenu();
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//function for bkground
void MENU::bkgrnd(int v)
 {
     clrscr();
    int   driver = DETECT,mode;
    int   x[15],y[15];
    int   x_center , y_center , rad = 90;
    int   i,j;

    initgraph(&driver,&mode,"c:\\tc\\bgi");
     x_center=getmaxx()/2;
     y_center=getmaxy()/2;

     setbkcolor(0);
     setcolor(BLUE);
     if(v==0)
       rad=100;
     if(v==1)
```

```
      rad=2000;
    for (;rad<=2000;rad+=50)
    {
        for ( i = 0; i < 15; i++ )
         {
             x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
             y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
         }
        for ( i = 0; i < 15; i++ )
         for ( j = 0; j < 15; j++ )
           {
              line(x[i],y[i],x[j],y[j]);
                if(v==1)
              delay(20);
           }

        if(rad==2000)
          break;

        cleardevice();
    }
}

int MENU::bkgrnd(void)
{
    clrscr();
   int   driver = DETECT,mode;
   int   x[15],y[15];
   int   x_center , y_center , rad = 2000;
   int   i,j,k;
   int   color=0,maxcolor,v=1;


   initgraph(&driver,&mode,"c:\\tc\\bgi");
    x_center=getmaxx()/2;
    y_center=getmaxy()/2;
    maxcolor=getmaxcolor();

    setbkcolor(0);
    while(k=(check(0,7))==9)
    {    setcolor(color);
        for ( i = 0; i < 15; i++ )
         {
             x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
             y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
         }
        for ( i = 0; i < 15; i++ )
         for ( j = 0; j < 15; j++ )
                line(x[i],y[i],x[j],y[j]);


        if(color==maxcolor)
           v=-1;
        if(color==0)
           v=1;

        delay(500);
        color+=v;
```

```
        // cleardevice();
     }
     return(k);
 }
int MENU::check(int n,int m)
{int i,flag,temp;
char ch[10];
gets(ch);
temp=atoi(ch);
for(i=n;i<=m;i++)
if(temp==i)
{flag=1;
break;}

if(flag==1)
return(temp);
else
return(9);
}
//function display edit menu
void MENU::editmenu(void)
     {
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          //clrscr();

          gotoxy(32,9);cout<<"EDIT MENU";
          gotoxy(32,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.BOOKS";
          gotoxy(34,14);cout<<"2.MEMBERS";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(31,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          editbook();
          else
          if(ch=='2')
          editmem();
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display edit menu for book and control
//all the func. in edit menu
void MENU::editbook(void)
{
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
```

```
          //clrscr();
          gotoxy(31,9);cout<<"EDIT BOOKS";
          gotoxy(31,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.MODIFY";
          gotoxy(34,14);cout<<"2.DELETE";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(31,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          {
               WORKING W;
               W.modifybook();
          }
          else
          if(ch=='2')
          {
               WORKING W;
               W.deletebook();
          }
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display edit menu for member and control
//all the func. in edit menu
void MENU::editmem(void)
{
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          clrscr();
          gotoxy(29,9);cout<<"EDIT MEMBERS";
          gotoxy(29,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.MODIFY";
          gotoxy(34,14);cout<<"2.DELETE";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(29,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          {
               WORKING W;
               W.modifymem();
          }
          else
          if(ch=='2')
          {
               WORKING W;
```

```
               W.deletemem();
          }
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display the introduction for the project
void MENU::introduc(void)
{
    clrscr();

    int   driver = DETECT,mode;
    int   x[15],y[15];
    int   x_center = 360, y_center = 180, rad = 90;
    int   i,j;
    char  *fname[]={"P","R" ,"O" ,"J","E","C","T","L","I","B","R","A","R","Y"};
    char  *fname2[]={"M","A" ,"D" ,"E "," ","B","Y","
","S","I","D","H","A","R","T","H","&"," ","N","I","K","H","I","L"," "};

    initgraph(&driver,&mode,"c:\\tc\\bgi");

    setbkcolor(0);
    setcolor(BLUE);
    delay(1000);
     for (x_center=110;x_center<=520;x_center+=10)
     {    for ( i = 0; i < 15; i++ )
          {
                x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
                y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
          }
         for ( i = 0; i < 15; i++ )
               for ( j = 0; j < 15; j++ )
                    {
                         line(x[i],y[i],x[j],y[j]);
                    }
               delay(100);
               if(x_center!=520)
               cleardevice();
     }
     setcolor(WHITE);
     settextstyle(4, HORIZ_DIR, 5);


     for(i=0;i<14;i++)
      {
           outtextxy(40+40*i,120+50*(i/7), *(fname+i));  /* output a message */
           delay(300);
      }
     setcolor(RED);
     settextstyle(0, HORIZ_DIR, 1);

     for(i=0;i<25;i++)
      {
           outtextxy(440+10*i-70*(i/16),440+12*(i/8), *(fname2+i));
           delay(100);
      }
```

```
     setcolor(BLUE);
     int  midx=getmaxx()/2;
       y_center=getmaxy()/2;
     for (x_center=520;x_center>=midx;x_center-=10)
       {    for ( i = 0; i < 15; i++ )
           {
                x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
                y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
           }
            for ( i = 0; i < 15; i++ )
              for ( j = 0; j < 15; j++ )
                 {
                  line(x[i],y[i],x[j],y[j]);
                 }
            delay(100);
            if(x_center!=midx)
            cleardevice();
       }
    for (rad=100;rad<=2500;rad+=50)
      {
            for ( i = 0; i < 15; i++ )
            {
                x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
                y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
            }
            for ( i = 0; i < 15; i++ )
            for ( j = 0; j < 15; j++ )
               {
               line(x[i],y[i],x[j],y[j]);
               }
            delay(100);
            if(rad==2000)
           break;
            cleardevice();
      }
   // getch();           /* press any key return to TEXT mode */
   // closegraph();
     gotoxy(31,5);cout<<"Welcome To Project";
     textcolor(RED+BLINK);textbackground(WHITE);
     gotoxy(33,7);cprintf("BOOK LIBRARY");
     textcolor(LIGHTGRAY);textbackground(BLACK);
     gotoxy(15,10);cout<<"This project has facility "<<"of maintaining records";
     gotoxy(15,11);cout<<"of BOOKS & MEMBERS.";
     gotoxy(15,13);cout<<"This project can hold "<<"more than 10,000 books";
     gotoxy(15,14);cout<<"records.";
     gotoxy(15,16);cout<<"One member can issue one"<<" book at a time.If he/she";
     gotoxy(15,17);cout<<"does not return book upto "<<"15 days he/she have to ";
     gotoxy(15,18);cout<<"pay fine of Rs.2/- per day.";
     textcolor(LIGHTGRAY+BLINK);
     gotoxy(27,24);cprintf("Press any key to continue");
     textcolor(LIGHTGRAY);
     getch();
     closegraph();
}
//this func. return 0 if given book code not found
int BOOK::bookfound(int tcode)
{
     fstream file;
```

```
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//return 0 if book name not found
int BOOK::booknamefound(char t1code[33])
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(!strcmpi(name,t1code))
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//return record no. for book code
int BOOK::recordno(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          count++;
          if(bookcode==tcode)break;
     }
     file.close();
     return count;
}
//returns available copies for given book code
int BOOK::available(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tavail=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tavail=avail;break;
          }
```

```
     }
     file.close();
     return tavail;
}
//return no. of copies for given book code
int BOOK::noofcopies(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tcopies=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tcopies=copies;break;
          }
     }
     file.close();
     return tcopies;
}
//return book name of the given book code
char *BOOK::bookname(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     char tname[33];
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               strcpy(tname,name);break;
          }
     }
     file.close();
     return tname;
}
//returns author name of the given book code
char *BOOK::authorname(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     char tauthor[26];
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               strcpy(tauthor,author);break;
          }
     }
     file.close();
     return tauthor;
}
//return book price of the given book code
float BOOK::bookprice(int tcode)
{
```

```
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     float tprice=0.0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tprice=price;break;
          }
     }
     file.close();
     return tprice;
}
//return book code of the given book name
int BOOK::bookcodeof(char t1code[33])
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tcode=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(!strcmpi(name,t1code))
          {
               tcode=bookcode;break;
          }
     }
     file.close();
     return tcode;
}
//return no. of records in the book file
int BOOK::reccount(void)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(BOOK)))
     count++;
     file.close();
     return count;
}
//deletes record of the given book code
void BOOK::deleterec(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(BOOK));
          if(file.eof())break;
          if(bookcode!=tcode)
          temp.write((char *)this,sizeof(BOOK));
     }
```

```
     file.close();
     temp.close();
     file.open("BOOK.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(BOOK));
          if(temp.eof())break;
          if(bookcode!=tcode)
          file.write((char *)this,sizeof(BOOK));
     }
     file.close();
     temp.close();
}
//add record in book file
void BOOK::addnewbook(int tcode,char tname[33],char tauthor[26],float tprice,int
tcopies,int tavail)
{
     fstream file;
     file.open("BOOK.DAT",ios::app);
     bookcode=tcode;
     strcpy(name,tname);
     strcpy(author,tauthor);
     price=tprice;
     copies=tcopies;
     avail=tavail;
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//updates the record in book file for given code
void BOOK::updatecopies(int tcode,int tcopies,int tavail)
{
     int recno;
     recno=recordno(tcode);
     fstream file;
     file.open("BOOK.DAT",ios::out | ios::ate);
     copies=tcopies;
     avail=tavail;
     int location;
     location=(recno-1)*sizeof(BOOK);
     file.seekp(location);
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//modify record in book file for given code
void BOOK::modify(int tcode,char tname[33],char tauthor[26],float tprice)
{
     int recno;
     recno=recordno(tcode);
     fstream file;
     file.open("BOOK.DAT",ios::out | ios::ate);
     strcpy(name,tname);
     strcpy(author,tauthor);
     price=tprice;
     int location;
     location=(recno-1)*sizeof(BOOK);
     file.seekp(location);
```

```
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//display record from book file for given book code
void BOOK::display(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               gotoxy(5,5);cout<<"Book Code :"<<bookcode;
               gotoxy(5,7);cout<<"Book Name :"<<name;
               gotoxy(5,8);cout<<"Author Name :"<<author;
               gotoxy(5,9);cout<<"Price :Rs. "<<price;
               gotoxy(5,10);cout<<"Copies   :"<<copies;
               gotoxy(5,11);cout<<"Available  :"<<avail;
               break;
          }
     }
     file.close();
}
//display list of books
void BOOK::list(void)
{
     clrscr();
     int row=6,found=0,flag=0;
     char ch;
     gotoxy(33,2);cout<<"  LIST OF BOOKS";
     gotoxy(32,3);cout<<"--------------------";
     gotoxy(1,4);cout<<"CODE      BOOK NAME                    AUTHOR
PRICE  COPIES";
     gotoxy(1,5);cout<<"------------------------------------------------------------
---------------------";
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(BOOK)))
     {
          flag=0;
          delay(20);
          found=1;
          gotoxy(2,row);cout<<bookcode;
          gotoxy(10,row);cout<<name;
          gotoxy(41,row);cout<<author;
          gotoxy(64,row);cout<<price;
          gotoxy(71,row);cout<<copies;
          textbackground(WHITE);textcolor(BLACK);
          gotoxy(40,row+1);cprintf("STATUS:");
          textcolor(BLACK+BLINK);
          cprintf("%d copies available",avail);
          textbackground(BLACK);textcolor(LIGHTGRAY);
          if(row==22)
          {
               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(33,2);cout<<"LIST OF BOOKS";
                gotoxy(32,3);cout<<"--------------------";
                gotoxy(1,4);cout<<"CODE        BOOK NAME          "
                          <<"         AUTHOR      PRICE  COPIES";
                gotoxy(1,5);
                cout<<"-------------------------------------------------"
                    <<"-------------------------------------------------";
           }
           else
           row=row+2;
      }
      if(!found)
      {
           gotoxy(5,10);cout<<"\t Records not found";
      }
      if(!flag)
      {
           gotoxy(1,25);cout<<"Press any key to continue...";
           getche();
      }
      file.close();
}
//returns 0 if given member code not found
int MEMBER::memfound(int mcode)
{
      fstream file;
      file.open("MEMBER.DAT",ios::in);
      file.seekg(0,ios::beg);
      int found=0;
      while(file.read((char *)this,sizeof(MEMBER)))
      {
           if(memcode==mcode)
           {
                found=1;break;
           }
      }
      file.close();
      return found;
}
//returns 0 if member have not issued any book
int MEMBER::issued(int mcode)
{
      fstream file;
      file.open("MEMBER.DAT",ios::in);
      file.seekg(0,ios::beg);
      int missue=0;
      while(file.read((char *)this,sizeof(MEMBER)))
      {
           if(memcode==mcode)
           {
                missue=bookcode;break;
           }
      }
```

```
     file.close();
     return missue;
}
//calculate & return fine for given mem. code
int MEMBER::fine(int mcode)
{
     DATE D;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int days,t_fine;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               days=D.diff(dd,mm,yy,d1,m1,y1);
               t_fine=days*2;
               break;
          }
     }
     file.close();
     return t_fine;
}
int MEMBER::lastcode(void)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int mcode=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     memcode=mcode;
     file.close();
     return mcode;
}
//returns mem. name given membercode
char *MEMBER::memname(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char mname[26];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(mname,name);break;
          }
     }
     file.close();
     return mname;
}
//returns mem. add. of given mem. code
```

```
char *MEMBER::memphone(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char mphone[10];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(mphone,phone);break;
          }
     }
     file.close();
     return mphone;
}
char *MEMBER::memaddress(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char maddress[33];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(maddress,address);break;
          }
     }
     file.close();
     return maddress;
}
//returns rec. no. of given mem.
int MEMBER::recordno(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          count++;
          if(memcode==mcode)break;

     }
     file.close();
     return count;
}

//returns record no. of given member code
void MEMBER::deleterec(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
```

```
     {
          file.read((char *)this,sizeof(MEMBER));
          if(file.eof())break;
          if(memcode!=mcode)
          temp.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
     file.open("MEMBER.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(MEMBER));
          if(temp.eof())break;
          file.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
}
//update record for given mem. code
void MEMBER::updatebook(int mcode,int tcode,int d1,int m1,int y1)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(MEMBER));
          if(file.eof())break;
          if(memcode==mcode)
          {
               bookcode=tcode;
               dd=d1;
               mm=m1;
               yy=y1;
               temp.write((char *)this,sizeof(MEMBER));
          }
          else
          temp.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
     file.open("MEMBER.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(MEMBER));
          if(temp.eof())break;
          file.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
}
//modify record for given mem. code
```

```
void MEMBER::modify(int mcode,char mname[26],char mphone[10],char maddress[33])
{
     int recno;
     recno=recordno(mcode);
     fstream file;
     file.open("MEMBER.DAT",ios::out | ios::ate);
     strcpy(name,mname);
     strcpy(phone,mphone);
     strcpy(address,maddress);
     int location;
     location=(recno-1)*sizeof(MEMBER);
     file.seekp(location);
     file.write((char *)this,sizeof(MEMBER));
     file.close();
}
//add rec. in file for given mem. code
void MEMBER::addmem(int mcode,int bcode,char mname[26],char maddress[33],char
mphone[10],int d1,int m1,int y1)
{
     fstream file;
     file.open("MEMBER.DAT",ios::app);
     memcode=mcode;
     bookcode=bcode;
     strcpy(name,mname);
     strcpy(address,maddress);
     strcpy(phone,mphone);
     dd=d1;
     mm=m1;
     yy=y1;
     file.write((char *)this,sizeof(MEMBER));
     file.close();
}
//display rec. for given mem. code
void MEMBER::display(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               gotoxy(5,3);cout<<"Member Code #"<<mcode;
               gotoxy(5,4);cout<<"----------------------";
               gotoxy(5,6);cout<<"Name  :"<<name;
               gotoxy(5,7);cout<<"Phone :"<<phone;
               gotoxy(5,8);cout<<"Address:"<<address;
               break;
          }
     }
     file.close();
}
//display list of all members
void MEMBER::list(void)
{
     clrscr();
     BOOK B;
     int row=6,found=0,flag=0;
```

```
     char ch;
     gotoxy(32,2);cout<<"LIST OF MEMBERS";
     gotoxy(31,3);cout<<"----------------------------";
     gotoxy(1,4);cout<<"CODE   BOOK CODE    NAME                          PHONE";
     gotoxy(1,5);cout<<"------------------------------------------------------------
---";
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          flag=0;
          delay(20);
          found=1;
          gotoxy(2,row);cout<<memcode;
          gotoxy(10,row);cout<<bookcode;
          gotoxy(19,row);cout<<name;
          gotoxy(48,row);cout<<phone;
          textbackground(WHITE);textcolor(BLACK);
          gotoxy(7,row+1);
          if(bookcode==0)
          cprintf("BOOK NAME:(Not Issued)");
          else
          {
               cprintf("BOOK NAME:%s",B.bookname(bookcode));
               gotoxy(42,row+1);
               cprintf("Date of Return:");
               textcolor(BLACK+BLINK);
               cprintf("%d%d%d",dd,mm,yy);
          }
          textbackground(BLACK);textcolor(LIGHTGRAY);
          if(row==22)
          {
               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(32,2);cout<<"LIST OF MEMBERS";
               gotoxy(31,3);cout<<"--------------------";
               gotoxy(1,4);cout<<"CODE         BOOK CODE         NAME
PHONE";
               gotoxy(1,5);
               cout<<"------------------------------------------------------------
--------"
                   <<"------------------------------------------------------------
--------";
          }
          else
          row=row+2;
     }
     if(!found)
     {
          gotoxy(5,10);cout<<"\7 Records not found";
     }
     if(!flag)
```

```
     {
          gotoxy(1,25);cout<<"Press any key to continue...";
          getche();
     }
     file.close();
}
//GIVES DATA TO ADD RECORD IN BOOK FILE
void WORKING::addbook(void)
{
     if(!reccount())          //mem. func. of BOOK
     {
          addnewbook(0,"null","null",0.0,0,0);
          BOOK::deleterec(0);
     }
     char ch;
     int tcode,tcopies,tavail;
     char tname[33],tauthor[26];
     float tprice=0.0;
     do
     {
          int found=0,valid=0;
          int tc;
          float t2=0.0;
          char t[10],t1[10];
          clrscr();
          gotoxy(29,3);cout<<"ADDITION OF THE BOOKS";
          gotoxy(29,4);cout<<"----------------------";
          gotoxy(72,1);cout<<"<0>=Exit";
          gotoxy(5,25);cout<<"Enter code no. of the book";
          gotoxy(5,5);cout<<"Code no.";
          gets(t);
          tc=atoi(t);
          tcode=tc;
          if(tcode==0)return;
          if(bookfound(tcode))
          {
               found=1;
               gotoxy(19,8);cout<<bookname(tcode);
               gotoxy(19,9);cout<<authorname(tcode);
               gotoxy(22,10);cout<<bookprice(tcode);
          }
          gotoxy(5,8);cout<<"Book Name :";
          gotoxy(5,9);cout<<"Author Name:";
          gotoxy(5,10);cout<<"Price   :Rs.";
          gotoxy(5,12);cout<<"Copies   :";
          valid=0;
          while(!valid && !found)
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter the name of the book";
               gotoxy(19,8);clreol();
               gets(tname);
               strupr(tname);
               if(tname[0]=='0')return;
               if(strlen(tname)<1 || strlen(tname)>32)
               {
                    valid=0;
```

```
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly (Range:1..32)";
          getch();
     }
}
valid=0;
while(!valid && !found)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the author's name of the book";
     gotoxy(19,9);clreol();
     gets(tauthor);
     strupr(tauthor);
     if(tauthor[0]=='0')return;
     if(strlen(tauthor)<1 || strlen(tauthor)>25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly (Range:1..25)";
          getch();
     }
}
valid=0;
while(!valid && !found)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the price of the book";
     gotoxy(22,10);clreol();
     gets(t1);
     t2=atof(t1);
     tprice=t2;
     if(t1[0]=='0')return;
     if(tprice<1 || tprice > 9999)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly";
          getch();
     }
}
valid=0;
while(!valid)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter no. of copies of book to be added";
     gotoxy(19,12);clreol();
     gets(t);
     tc=atoi(t);
     tcopies=tc;
     if(t[0]=='0')return;
     if(tcopies<1 || tcopies > 50)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly";
```

```
                    getch();
               }
          }
          tavail=available(tcode) + tcopies;
          tcopies=noofcopies(tcode) + tcopies;
          gotoxy(5,25);clreol();
          do
          {
               gotoxy(5,15);clreol();
               cout<<"Do you want to save(y/n):";
               ch=getche();
               ch=toupper(ch);
          }while(ch != 'Y' && ch !='N');
          if(ch=='Y')
          {
               if(found)
               updatecopies(tcode,tcopies,tavail);
               else
               addnewbook(tcode,tname,tauthor,tprice,tcopies,tavail);
          }
          do
          {
               gotoxy(5,17);clreol();
               cout<<"Do you want to add more(y/n):";
               ch=getche();
               ch=toupper(ch);
          }while(ch != 'Y' && ch != 'N');
     }while(ch=='Y');
}
//gives data to add record in mem. file
void WORKING::addmember(void)
{
     char ch;
     int mcode,bcode;
     char mname[26],mphone[10],maddress[33];
     int d1,m1,y1;
     mcode=lastcode();
     mcode++;
     do
     {
          int valid=0;
          clrscr();
          gotoxy(28,3);
          cout<<"ADDITION OF THE MEMBERS";
          gotoxy(28,4);
          cout<<"--------------------------";
          gotoxy(72,1);
          cout<<"<0>=Exit";
          gotoxy(5,7);
          cout<<"Member Code #"<<mcode;
          gotoxy(5,8);
          cout<<"-----------------";
          gotoxy(5,10);cout<<"Name :";
          gotoxy(5,12);cout<<"Phone :";
          gotoxy(5,14);cout<<"Address:";
          do
          {
               valid=1;
```

```
                gotoxy(5,25);clreol();
                cout<<"Enter the name of New Member";
                gotoxy(15,10);clreol();
                gets(mname);
                strupr(mname);
                if(mname[0]=='0')
                return;
                if(strlen(mname)<1 || strlen(mname) > 25)
                {
                     valid=0;
                     gotoxy(5,25);clreol();
                     cout<<"\7Enter correctly(Range: 1..25)";
                     getch();
                }
           }while(!valid);
           do
           {
                valid=1;
                gotoxy(5,25);clreol();
                cout<<"Enter Phone no. of the Member or Press <ENTER> for none";
                gotoxy(15,12);clreol();
                gets(mphone);
                if(mphone[0]=='0')return;
                if((strlen(mphone) < 7 && strlen(mphone) > 0) || (strlen(mphone) >
9))
                {
                     valid=0;
                     gotoxy(5,25);clreol();
                     cout<<"\7Enter correctly";
                     getch();
                }
           }while(!valid);
           if(strlen(mphone)==0)strcpy(mphone,"-");
           do
           {
                valid=1;
                gotoxy(5,25);clreol();
                cout<<"Enter the address of the New Member";
                gotoxy(15,14);clreol();
                gets(maddress);
                strupr(maddress);
                if(maddress[0]=='0')return;
                if(strlen(maddress)<1 || strlen(maddress)>32)
                {
                     valid=0;
                     gotoxy(5,25);clreol();
                     cout<<"\7Enter correctly(Range: 1..32)";
                     getch();
                }
           }while(!valid);
           gotoxy(5,25);clreol();
           do
           {
                gotoxy(5,17);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=='Y')
          {
               bcode=0;
               d1=0;
               m1=0;y1=0;
               addmem(mcode,bcode,mname,maddress,mphone,d1,m1,y1);
               mcode++;
          }
          do
          {
               gotoxy(5,19);clreol();
               cout<<"Do you want to add more(y/n):";
               ch=getche();
               ch=toupper(ch);
               if(ch=='0')return;
          }while(ch!='Y' && ch!='N');
     }while(ch=='Y');
}
//issues the book
void WORKING::issuebook(void)
{
     BOOK B;
     MEMBER M;
     DATE D;
     char t1code[33],ch;
     int t2code=0,tcode=0,mcode=0;
     int valid;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be issued";
               gotoxy(5,6);cout<<"             or                 ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.........:";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
```

```
                gotoxy(5,10);cout<<"\7Record not found";
                gotoxy(5,11);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
      }while(!valid);
      if(tcode==0)
      tcode=bookcodeof(t1code);
      if(!available(tcode))
      {
           gotoxy(5,10);
           cout<<"\7Sorry! Book("<<bookname(tcode)<<") is not available";
           gotoxy(5,11);cout<<"Kindly issue another book";
           gotoxy(5,12);
           cout<<"See list of Books";
           getch();
           return;
      }
      do
      {
           valid=1;
           while(1)
           {
                clrscr();
                gotoxy(72,1);cout<<"<0>=Exit";
                gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
                gotoxy(5,5);cout<<"Book Name:"<<bookname(tcode);
                gotoxy(5,7);cout<<"Enter Code no. of the Member";
                gotoxy(5,8);cout<<"             or                 ";
                gotoxy(5,9);cout<<"Press <ENTER> for help.......: ";
                gets(t1code);
                if(t1code[0]=='0')return;
                if(strlen(t1code)==0)
                M.list();
                else
                break;
           }
           t2code=atoi(t1code);
           mcode=t2code;
           if(mcode==0)
           {
                valid=0;
                gotoxy(5,25);
                cout<<"\7Enter Correctly";
                getch();
           }
           if(!memfound(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Record not found";
                gotoxy(5,14);cout<<"Press <ESC> to exit or any other Key to
continue...";
                ch=getch();
                if(ch==27)return;
           }
           if(issued(mcode) && valid)
           {
```

```
                valid=0;
                gotoxy(5,13);cout<<"\7Sorry!! you can not issue more than one";
                gotoxy(5,14);cout<<"Press <ESC> to exit or any other Key to
continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     int tcopies,tavail;
     tcopies=noofcopies(tcode);     //mem. func. of BOOK
     tavail=available(tcode)-1;
     updatecopies(tcode,tcopies,tavail);
     D.extenddate(d1,m1,y1,15);
     d1=D.day;
     m1=D.mon;
     y1=D.year;
     updatebook(mcode,tcode,d1,m1,y1);     //"      " " MEMBER
     gotoxy(5,13);
     cout<<"\7Book is issued to "<<memname(mcode);
     gotoxy(5,15);
     cout<<"Date of Return: "<<d1<<"/"<<m1<<"/"<<y1;
     getch();
}
void WORKING::returnbook(void)
{
     MEMBER M;
     char t1code[5],ch;
     int t2code=0,mcode=0;
     int valid;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(5,7);cout<<"Enter Code of the Member who wants to return
Book";
               gotoxy(5,8);cout<<"             or                 ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               M.list();
               else
               break;
          }
          t2code=atoi(t1code);
          mcode=t2code;
          if(mcode==0)
          {
```

```
                valid=0;
                gotoxy(5,25);cout<<"\7Enter Correctly";
                getch();
           }
           if(!memfound(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Record not found";
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
           if(!issued(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Member have no book to return";
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
      }while(!valid);
      int bcode,tcopies,tavail;
      bcode=issued(mcode);
      gotoxy(5,13);cout<<"Book Code:"<<bcode;
      gotoxy(5,14);cout<<"Book Name:"<<bookname(bcode);
      tcopies=noofcopies(bcode);
      tavail=available(bcode)+1;
      int f;
      f=fine(mcode);
      if(f!=0)
      {
           gotoxy(5,16);cout<<"You have to pay a fine of Rs."<<f;
           gotoxy(5,17);cout<<"Please do not delay the Return of book again";
      }
      updatecopies(bcode,tcopies,tavail);
      updatebook(mcode,0,0,0,0);
      gotoxy(5,19);cout<<"\7Book has been returned";
      getch();
}
//gives data to modify book record
void WORKING::modifybook(void)
{
      BOOK B;
      char t1code[5],tname[33],tauthor[26],*t1,ch;
      int t2code=0,tcode=0;
      float t2=0.0,tprice=0.0;
      int valid;
      do
      {
           valid=1;
           while(1)
           {
                clrscr();
                gotoxy(72,1);cout<<"<0>=Exit";
                gotoxy(5,5);cout<<"Enter Code or Name of the Book to be modified";
                gotoxy(5,6);cout<<"             or                 ";
```

```
               gotoxy(5,7);cout<<"Press <ENTER> for help.........: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
     tcode=bookcodeof(t1code);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     BOOK::display(tcode);
     do
     {
          gotoxy(5,13);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;
     gotoxy(5,16);cout<<"Book Name :";
     gotoxy(5,17);cout<<"Author Name:";
     gotoxy(5,18);cout<<"Price   :Rs.";
     do
     {
          valid=1;
          gotoxy(5,25);clreol();
          cout<<"Enter the name of the book or <ENTER> for no change";
          gotoxy(19,16);clreol();
          gets(tname);
          strupr(tname);
          if(tname[0]=='0')return;
          if(strlen(tname)>32)
          {
               valid=0;
               gotoxy(5,25);clreol();
               cout<<"\7Enter correctly(Range: 1..32)";
               getch();
           }
      }while(!valid);
      if(strlen(tname)==0)
      strcpy(tname,bookname(tcode));
      do
```

```
      {
          valid=1;
          gotoxy(5,25);clreol();
          cout<<"Enter the author's name or <ENTER> for no change";
          gotoxy(19,17);clreol();
          gets(tauthor);
          strupr(tauthor);
          if(tauthor[0]=='0')return;
          if(strlen(tauthor)>25)
          {
               valid=0;
               gotoxy(5,25);clreol();
               cout<<"\7Enter Correctly(Range: 1..25)";
               getch();
          }
      }while(!valid);
      if(strlen(tauthor)==0)
      strcpy(tauthor,authorname(tcode));
      do
      {
          valid=1;
          gotoxy(5,25);clreol();
          cout<<"Enter price or <ENTER> for no change";
          gotoxy(22,18);clreol();
          gets(t1);
          t2=atof(t1);
          tprice=t2;
          if(t1[0]=='0')return;
          if((tprice<1 || tprice > 9999) && (t1[0]!='\0'))
          {
               valid=0;
               gotoxy(5,25);clreol();
               cout<<"\7 Enter correctly";
               getch();
          }
     }while(!valid);
     if(strlen(t1)==0)
     tprice=bookprice(tcode);
     gotoxy(5,25);
     clreol();
     do
     {
          gotoxy(5,20);clreol();
          cout<<"Do you want to save changes(y/n)";
          ch=getche();
          ch=toupper(ch);
          if(ch=='0')return;
     }while(ch!='Y' && ch!='N');
     if(ch=='N')
     return;
     BOOK::modify(tcode,tname,tauthor,tprice);
     gotoxy(5,23);cout<<"\7Record Modified";
     getch();
}
//gives data to modify member record
void WORKING::modifymem(void)
{
     MEMBER M;
```

```
char m1code[10],mname[26],mphone[10],maddress[33],ch;
int m2code=0,mcode=0;
int valid;
do
{
     valid=1;
     while(1)
     {
          clrscr();
          gotoxy(72,1);cout<<"<0>=Exit";
          gotoxy(5,7);cout<<"Enter Code no. of the Member to be modified";
          gotoxy(5,8);cout<<"             or                 ";
          gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
          gets(m1code);
          m2code=atoi(m1code);
          mcode=m2code;
          if(m1code[0]=='0')return;
          if(strlen(m1code)==0)
          M.list();
          else
          break;
     }
     if(mcode==0)
     {
          valid=0;
          gotoxy(5,25);cout<<"\7Enter Correctly";
          getch();
     }
     if(valid && !memfound(mcode))
     {
          valid=0;
          gotoxy(5,13);cout<<"\7Record not found";
          gotoxy(5,14);
          cout<<"Press <ESC> to exit or any other key to continue...";
          ch=getch();
          if(ch==27)return;
     }
}while(!valid);
clrscr();
gotoxy(72,1);cout<<"<0>=Exit";
MEMBER::display(mcode);
do
{
     gotoxy(5,10);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;
gotoxy(5,13);cout<<"Name  :";
gotoxy(5,14);cout<<"Phone  :";
gotoxy(5,15);cout<<"Address:";
do
{
     valid=1;
     gotoxy(5,25);clreol();
```

```
     cout<<"Enter the name of the member or <ENTER> for no change";
     gotoxy(19,13);clreol();
     gets(mname);
     strupr(mname);
     if(mname[0]=='0')return;
     if(strlen(mname) > 25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly(Range: 1..25)";
          getch();
     }
}while(!valid);
if(strlen(mname)==0)
strcpy(mname,memname(mcode));
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the Phone no. of Member or <ENTER> for no change";
     gotoxy(19,14);clreol();
     gets(mphone);
     if(mphone[0]=='0')return;
     if((strlen(mphone) < 7 && strlen(mphone) > 0 || (strlen(mphone) > 9)))
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter Correctly";
          getch();
     }
}while(!valid);
if(strlen(mphone)==0)
strcpy(mphone,memphone(mcode));
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the address of the member or <ENTER> for no change";
     gotoxy(19,15);clreol();
     gets(maddress);
     strupr(maddress);
     if(maddress[0]=='0')return;
     if(strlen(maddress) > 32)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly(Range: 1..32)";
          getch();
     }
}while(!valid);
if(strlen(maddress)==0)
strcpy(maddress,memaddress(mcode));
gotoxy(5,25);clreol();
do
{
     gotoxy(5,18);clreol();
     cout<<"Do you want to save changes(y/n)";
     ch=getche();
```

```
          ch=toupper(ch);
          if(ch=='0')return;
     }while(ch!='Y' && ch!='N');
     if(ch=='N')
     return;
     MEMBER::modify(mcode,mname,mphone,maddress);
     gotoxy(5,23);
     cout<<"\7Record Modified";
     getch();
}
//gives book code to delete book record
void WORKING::deletebook(void)
{
     BOOK B;
     char t1code[5],tname[33],tauthor[26],ch;
     int t2code=0,tcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be deleted";
               gotoxy(5,6);cout<<"             or                  ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.........: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
     tcode=bookcodeof(t1code);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     BOOK::display(tcode);
     do
     {
          gotoxy(5,13);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;
     int tavail,tcopies;
     tavail=available(tcode);
     tcopies=noofcopies(tcode);
     if(tavail!=tcopies)
     {
          gotoxy(5,15);cout<<"\7Record cannot be deleted.This book is issued.";
          gotoxy(5,16);cout<<"\7Type special code to delete";
          if(getch()!='7')
          {        gotoxy(5,19);cout<<"\7*";
                getch();
                return;
          }
          else
                gotoxy(5,19);cout<<"\7*";
     }
     BOOK::deleterec(tcode);
     gotoxy(5,23);cout<<"\7Record Deleted";
     getch();
}
//gives mem. code to delete member record
void WORKING::deletemem(void)
{
     MEMBER M;
     char m1code[10],mname[26],mphone[10],maddress[33],ch;
     int m2code=0,mcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,7);cout<<"Enter Code no. of the Member to be deleted";
               gotoxy(5,8);cout<<"             or                  ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
               gets(m1code);
               m2code=atoi(m1code);
               mcode=m2code;
               if(m1code[0]=='0')return;
               if(strlen(m1code)==0)
               M.list();
               else
               break;
          }
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);cout<<"\7Enter Correctly";
               getch();
          }
          if(valid && !memfound(mcode))
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
```

```
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
          }
     }while(!valid);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     MEMBER::display(mcode);
     do
     {
          gotoxy(5,10);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;
     if(issued(mcode))
     {
          gotoxy(5,15);
          cout<<"\7Record cannot be deleted.Member has a book.";
          getch();
          return;
     }
     MEMBER::deleterec(mcode);
     gotoxy(5,23);cout<<"\7Record Modified";
     getch();
}
//Main func. calling introduction and main menu
void main()
{
     MENU menu;
     menu.introduc();
     menu.mainmenu();
}
```

```
#include<iostream.h>
#include<process.h>
#include<string.h>
#include<stdlib.h>
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<dos.h>
#include<fstream.h>
#include<graphics.h>
#include<math.h>
class MENU
{
     public:
          void mainmenu(void);
          void introduc(void);
          void bkgrnd(int);
          int bkgrnd(void);
          int check(int,int);
     private:
          void editmenu(void);
          void editbook(void);
          void editmem(void);
};
class BOOK
{
     public:
          void list(void);
          char *bookname(int);
     protected:
          void addnewbook(int,char tname[33],char tauthor[26],float,int,int);
          void updatecopies(int,int,int);
          void modify(int,char[],char[],float);
          void deletion(void);
          int bookfound(int);
          int booknamefound(char[]);
          int recordno(int);
          int available(int);
          char *authorname(int);
          float bookprice(int);
          int noofcopies(int);
          int bookcodeof(char[]);
          void display(int);
          int reccount(void);
          void deleterec(int);
     private:
          int bookcode,copies;
          char name[33],author[26];
```

```
          float price;
          int avail;
};
class MEMBER
{
     public:
          void list(void);
     protected:
          void addmem(int,int,char[],char[],char[],int,int,int);
          void modify(int,char[],char[],char[]);
          void deletion(void);
          int memfound(int);
          void updatebook(int,int,int,int,int);
          char *memname(int);
          char *memphone(int);
          char *memaddress(int);
          int recordno(int);
          int lastcode(void);
          int issued(int);
          int fine(int);
          void display(int);
          void deleterec(int);
     private:
          int memcode,bookcode;
          char name[26],phone[10],address[33];
          int dd,mm,yy;
};
class WORKING:public BOOK,public MEMBER
{
     public:
          void issuebook(void);
          void returnbook(void);
          void addbook(void);
          void addmember(void);
          void modifybook(void);
          void modifymem(void);
          void deletebook(void);
          void deletemem(void);
};
class DATE
{
     public:
          void extenddate(int,int,int,int);
          int diff(int,int,int,int,int,int);
          int day,mon,year;
};
void DATE::extenddate(int d1,int m1,int y1,int days)
{
     static int month[]={31,29,31,30,31,30,31,31,30,31,30,31};
     for(int i=1;i<=days;i++)
     {
          d1++;
          if((d1>month[m1-1]) || (y1%4!=0 && m1==2 && d1>28))
          {
               d1=1;
               m1++;
          }
          if(m1>12)
```

```
          {
               m1=1;
               y1++;
          }
     }
     day=d1;
     mon=m1;
     year=y1;
}
int DATE::diff(int d1,int m1,int y1,int d2,int m2,int y2)
{
     int days=0;
     if((y2<y1) || (y2==y1 && m2<m1) || (y2==y1 && m2==m1 && d2<d1))
     return days;
     static int month[]={31,29,31,30,31,30,31,31,30,31,30,31};
     while(d1!=d2 || m1!=m2 || y1!=y2)
     {
          days++;
          d1++;
          if((d1>month[m1-1]) || (y1%4!=0 && m1==2 && d1>28))
          {
               d1=1;
               m1++;
          };
          if(m1>12)
          {
               m1=1;
               y1++;
          }
     }
     return days;
}
void MENU::mainmenu(void)
{

     char ch;
     int p;
     MENU::bkgrnd(0);
     while(1)
     {
           // cleardevice();


          gotoxy(29,6);
          cout<<"BOOK LIBRARY";
          gotoxy(29,7);
          cout<<"------------------------------";
          gotoxy(30,10);cout<<"1.Introduction";
          gotoxy(30,11);cout<<"2.Add New Book(s)";
          gotoxy(30,12);cout<<"3.Add New Member(s)";
          gotoxy(30,13);cout<<"4.Issue Book(s)";
          gotoxy(30,14);cout<<"5.Return Book(s)";
          gotoxy(30,15);cout<<"6.List of Book(s)";
          gotoxy(30,16);cout<<"7.List of Member(s)";
          gotoxy(30,17);cout<<"8.EDIT";
          gotoxy(30,18);cout<<"0.QUIT";
          gotoxy(30,20);cout<<"Enter Your Choice:";
          ch=getch();
```

```
          closegraph();
          if(ch==27)
          break;
          if(ch=='1')
          introduc();
          else
          if(ch=='2')
          {
               WORKING W;
               W.addbook();
          }
          else
          if(ch=='3')
          {
               WORKING W;
               W.addmember();
          }
          else
          if(ch=='4')
          {
               WORKING W;
               W.issuebook();
          }
          else
          if(ch=='5')
          {
               WORKING W;
               W.returnbook();
          }
          else
          if(ch=='6')
          {
               BOOK B;
               B.list();
          }
          else
          if(ch=='7')
          {
               MEMBER M;
               M.list();
          }
          else
          if(ch=='8')
          editmenu();
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//function for bkground
void MENU::bkgrnd(int v)
 {
     clrscr();
    int   driver = DETECT,mode;
    int   x[15],y[15];
    int   x_center , y_center , rad = 90;
    int   i,j;
```

```
   initgraph(&driver,&mode,"c:\\tc\\bgi");
    x_center=getmaxx()/2;
    y_center=getmaxy()/2;

    setbkcolor(0);
    setcolor(BLUE);
    if(v==0)
      rad=100;
    if(v==1)
      rad=2000;
    for (;rad<=2000;rad+=50)
    {
        for ( i = 0; i < 15; i++ )
         {
             x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
             y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
         }
        for ( i = 0; i < 15; i++ )
         for ( j = 0; j < 15; j++ )
           {
              line(x[i],y[i],x[j],y[j]);
                if(v==1)
              delay(20);
           }

        if(rad==2000)
          break;

        cleardevice();
    }
}

int MENU::bkgrnd(void)
{
    clrscr();
   int   driver = DETECT,mode;
   int   x[15],y[15];
   int   x_center , y_center , rad = 2000;
   int   i,j,k;
   int   color=0,maxcolor,v=1;


   initgraph(&driver,&mode,"c:\\tc\\bgi");
    x_center=getmaxx()/2;
    y_center=getmaxy()/2;
    maxcolor=getmaxcolor();

    setbkcolor(0);
    while(k=(check(0,7))==9)
    {    setcolor(color);
        for ( i = 0; i < 15; i++ )
         {
             x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
             y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
         }
        for ( i = 0; i < 15; i++ )
         for ( j = 0; j < 15; j++ )
```

```
                line(x[i],y[i],x[j],y[j]);


         if(color==maxcolor)
            v=-1;
         if(color==0)
            v=1;

         delay(500);
         color+=v;
        // cleardevice();
     }
     return(k);
 }
int MENU::check(int n,int m)
{int i,flag,temp;
char ch[10];
gets(ch);
temp=atoi(ch);
for(i=n;i<=m;i++)
if(temp==i)
{flag=1;
break;}

if(flag==1)
return(temp);
else
return(9);
}
//function display edit menu
void MENU::editmenu(void)
     {
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          //clrscr();

          gotoxy(32,9);cout<<"EDIT MENU";
          gotoxy(32,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.BOOKS";
          gotoxy(34,14);cout<<"2.MEMBERS";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(31,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          editbook();
          else
          if(ch=='2')
          editmem();
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
```

```
     }
}
//func. to display edit menu for book and control
//all the func. in edit menu
void MENU::editbook(void)
{
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          //clrscr();
          gotoxy(31,9);cout<<"EDIT BOOKS";
          gotoxy(31,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.MODIFY";
          gotoxy(34,14);cout<<"2.DELETE";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(31,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          {
               WORKING W;
               W.modifybook();
          }
          else
          if(ch=='2')
          {
               WORKING W;
               W.deletebook();
          }
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display edit menu for member and control
//all the func. in edit menu
void MENU::editmem(void)
{
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          clrscr();
          gotoxy(29,9);cout<<"EDIT MEMBERS";
          gotoxy(29,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.MODIFY";
          gotoxy(34,14);cout<<"2.DELETE";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(29,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
```

```
          else
          if(ch=='1')
          {
               WORKING W;
               W.modifymem();
          }
          else
          if(ch=='2')
          {
               WORKING W;
               W.deletemem();
          }
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display the introduction for the project
void MENU::introduc(void)
{
    clrscr();

    int   driver = DETECT,mode;
    int   x[15],y[15];
    int   x_center = 360, y_center = 180, rad = 90;
    int   i,j;
    char  *fname[]={"P","R" ,"O" ,"J","E","C","T","L","I","B","R","A","R","Y"};
    char  *fname2[]={"M","A" ,"D" ,"E "," ","B","Y","
","S","I","D","H","A","R","T","H","&"," ","N","I","K","H","I","L"," "};

    initgraph(&driver,&mode,"c:\\tc\\bgi");

    setbkcolor(0);
    setcolor(BLUE);
    delay(1000);
     for (x_center=110;x_center<=520;x_center+=10)
     {    for ( i = 0; i < 15; i++ )
          {
                x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
                y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
          }
         for ( i = 0; i < 15; i++ )
               for ( j = 0; j < 15; j++ )
                    {
                         line(x[i],y[i],x[j],y[j]);
                    }
               delay(100);
               if(x_center!=520)
               cleardevice();
     }
     setcolor(WHITE);
     settextstyle(4, HORIZ_DIR, 5);


     for(i=0;i<14;i++)
      {
           outtextxy(40+40*i,120+50*(i/7), *(fname+i));  /* output a message */
```

```
        delay(300);
     }
    setcolor(RED);
    settextstyle(0, HORIZ_DIR, 1);

    for(i=0;i<25;i++)
     {
        outtextxy(440+10*i-70*(i/16),440+12*(i/8), *(fname2+i));
        delay(100);
     }
    setcolor(BLUE);
    int  midx=getmaxx()/2;
      y_center=getmaxy()/2;
    for (x_center=520;x_center>=midx;x_center-=10)
      {    for ( i = 0; i < 15; i++ )
          {
              x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
              y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
          }
           for ( i = 0; i < 15; i++ )
             for ( j = 0; j < 15; j++ )
                {
                 line(x[i],y[i],x[j],y[j]);
                }
           delay(100);
           if(x_center!=midx)
           cleardevice();
      }
   for (rad=100;rad<=2500;rad+=50)
     {
           for ( i = 0; i < 15; i++ )
           {
               x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
               y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
           }
           for ( i = 0; i < 15; i++ )
           for ( j = 0; j < 15; j++ )
              {
              line(x[i],y[i],x[j],y[j]);
              }
           delay(100);
           if(rad==2000)
          break;
           cleardevice();
     }
  // getch();           /* press any key return to TEXT mode */
  // closegraph();
    gotoxy(31,5);cout<<"Welcome To Project";
    textcolor(RED+BLINK);textbackground(WHITE);
    gotoxy(33,7);cprintf("BOOK LIBRARY");
    textcolor(LIGHTGRAY);textbackground(BLACK);
    gotoxy(15,10);cout<<"This project has facility "<<"of maintaining records";
    gotoxy(15,11);cout<<"of BOOKS & MEMBERS.";
    gotoxy(15,13);cout<<"This project can hold "<<"more than 10,000 books";
    gotoxy(15,14);cout<<"records.";
    gotoxy(15,16);cout<<"One member can issue one"<<" book at a time.If he/she";
    gotoxy(15,17);cout<<"does not return book upto "<<"15 days he/she have to ";
    gotoxy(15,18);cout<<"pay fine of Rs.2/- per day.";
```

```
     textcolor(LIGHTGRAY+BLINK);
     gotoxy(27,24);cprintf("Press any key to continue");
     textcolor(LIGHTGRAY);
     getch();
     closegraph();
}
//this func. return 0 if given book code not found
int BOOK::bookfound(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//return 0 if book name not found
int BOOK::booknamefound(char t1code[33])
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(!strcmpi(name,t1code))
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//return record no. for book code
int BOOK::recordno(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          count++;
          if(bookcode==tcode)break;
     }
     file.close();
     return count;
}
//returns available copies for given book code
int BOOK::available(int tcode)
{
```

```
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tavail=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tavail=avail;break;
          }
     }
     file.close();
     return tavail;
}
//return no. of copies for given book code
int BOOK::noofcopies(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tcopies=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tcopies=copies;break;
          }
     }
     file.close();
     return tcopies;
}
//return book name of the given book code
char *BOOK::bookname(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     char tname[33];
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               strcpy(tname,name);break;
          }
     }
     file.close();
     return tname;
}
//returns author name of the given book code
char *BOOK::authorname(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     char tauthor[26];
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
```

```
          {
               strcpy(tauthor,author);break;
          }
     }
     file.close();
     return tauthor;
}
//return book price of the given book code
float BOOK::bookprice(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     float tprice=0.0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tprice=price;break;
          }
     }
     file.close();
     return tprice;
}
//return book code of the given book name
int BOOK::bookcodeof(char t1code[33])
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tcode=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(!strcmpi(name,t1code))
          {
               tcode=bookcode;break;
          }
     }
     file.close();
     return tcode;
}
//return no. of records in the book file
int BOOK::reccount(void)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(BOOK)))
     count++;
     file.close();
     return count;
}
//deletes record of the given book code
void BOOK::deleterec(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
```

```
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(BOOK));
          if(file.eof())break;
          if(bookcode!=tcode)
          temp.write((char *)this,sizeof(BOOK));
     }
     file.close();
     temp.close();
     file.open("BOOK.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(BOOK));
          if(temp.eof())break;
          if(bookcode!=tcode)
          file.write((char *)this,sizeof(BOOK));
     }
     file.close();
     temp.close();
}
//add record in book file
void BOOK::addnewbook(int tcode,char tname[33],char tauthor[26],float tprice,int
tcopies,int tavail)
{
     fstream file;
     file.open("BOOK.DAT",ios::app);
     bookcode=tcode;
     strcpy(name,tname);
     strcpy(author,tauthor);
     price=tprice;
     copies=tcopies;
     avail=tavail;
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//updates the record in book file for given code
void BOOK::updatecopies(int tcode,int tcopies,int tavail)
{
     int recno;
     recno=recordno(tcode);
     fstream file;
     file.open("BOOK.DAT",ios::out | ios::ate);
     copies=tcopies;
     avail=tavail;
     int location;
     location=(recno-1)*sizeof(BOOK);
     file.seekp(location);
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//modify record in book file for given code
void BOOK::modify(int tcode,char tname[33],char tauthor[26],float tprice)
{
```

```
     int recno;
     recno=recordno(tcode);
     fstream file;
     file.open("BOOK.DAT",ios::out | ios::ate);
     strcpy(name,tname);
     strcpy(author,tauthor);
     price=tprice;
     int location;
     location=(recno-1)*sizeof(BOOK);
     file.seekp(location);
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//display record from book file for given book code
void BOOK::display(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               gotoxy(5,5);cout<<"Book Code :"<<bookcode;
               gotoxy(5,7);cout<<"Book Name :"<<name;
               gotoxy(5,8);cout<<"Author Name :"<<author;
               gotoxy(5,9);cout<<"Price :Rs. "<<price;
               gotoxy(5,10);cout<<"Copies   :"<<copies;
               gotoxy(5,11);cout<<"Available  :"<<avail;
               break;
          }
     }
     file.close();
}
//display list of books
void BOOK::list(void)
{
     clrscr();
     int row=6,found=0,flag=0;
     char ch;
     gotoxy(33,2);cout<<"  LIST OF BOOKS";
     gotoxy(32,3);cout<<"------------------";
     gotoxy(1,4);cout<<"CODE      BOOK NAME                    AUTHOR
PRICE  COPIES";
     gotoxy(1,5);cout<<"------------------------------------------------------------
---------------------";
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(BOOK)))
     {
          flag=0;
          delay(20);
          found=1;
          gotoxy(2,row);cout<<bookcode;
          gotoxy(10,row);cout<<name;
          gotoxy(41,row);cout<<author;
          gotoxy(64,row);cout<<price;
```

```
          gotoxy(71,row);cout<<copies;
          textbackground(WHITE);textcolor(BLACK);
          gotoxy(40,row+1);cprintf("STATUS:");
          textcolor(BLACK+BLINK);
          cprintf("%d copies available",avail);
          textbackground(BLACK);textcolor(LIGHTGRAY);
          if(row==22)
          {
               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(33,2);cout<<"LIST OF BOOKS";
               gotoxy(32,3);cout<<"--------------------";
               gotoxy(1,4);cout<<"CODE        BOOK NAME          "
                         <<"         AUTHOR      PRICE  COPIES";
               gotoxy(1,5);
               cout<<"-------------------------------------------------"
                   <<"-------------------------------------------------";
          }
          else
          row=row+2;
     }
     if(!found)
     {
          gotoxy(5,10);cout<<"\t Records not found";
     }
     if(!flag)
     {
          gotoxy(1,25);cout<<"Press any key to continue...";
          getche();
     }
     file.close();
}
//returns 0 if given member code not found
int MEMBER::memfound(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//returns 0 if member have not issued any book
int MEMBER::issued(int mcode)
{
     fstream file;
```

```
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int missue=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               missue=bookcode;break;
          }
     }
     file.close();
     return missue;
}
//calculate & return fine for given mem. code
int MEMBER::fine(int mcode)
{
     DATE D;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int days,t_fine;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               days=D.diff(dd,mm,yy,d1,m1,y1);
               t_fine=days*2;
               break;
          }
     }
     file.close();
     return t_fine;
}
int MEMBER::lastcode(void)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int mcode=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     memcode=mcode;
     file.close();
     return mcode;
}
//returns mem. name given membercode
char *MEMBER::memname(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char mname[26];
     while(file.read((char *)this,sizeof(MEMBER)))
```

```
     {
          if(memcode==mcode)
          {
               strcpy(mname,name);break;
          }
     }
     file.close();
     return mname;
}
//returns mem. add. of given mem. code
char *MEMBER::memphone(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char mphone[10];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(mphone,phone);break;
          }
     }
     file.close();
     return mphone;
}
char *MEMBER::memaddress(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char maddress[33];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(maddress,address);break;
          }
     }
     file.close();
     return maddress;
}
//returns rec. no. of given mem.
int MEMBER::recordno(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          count++;
          if(memcode==mcode)break;

     }
     file.close();
     return count;
}
```

```
//returns record no. of given member code
void MEMBER::deleterec(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(MEMBER));
          if(file.eof())break;
          if(memcode!=mcode)
          temp.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
     file.open("MEMBER.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(MEMBER));
          if(temp.eof())break;
          file.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
}
//update record for given mem. code
void MEMBER::updatebook(int mcode,int tcode,int d1,int m1,int y1)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(MEMBER));
          if(file.eof())break;
          if(memcode==mcode)
          {
               bookcode=tcode;
               dd=d1;
               mm=m1;
               yy=y1;
               temp.write((char *)this,sizeof(MEMBER));
          }
          else
          temp.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
     file.open("MEMBER.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
```

```
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(MEMBER));
          if(temp.eof())break;
          file.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
}
//modify record for given mem. code
void MEMBER::modify(int mcode,char mname[26],char mphone[10],char maddress[33])
{
     int recno;
     recno=recordno(mcode);
     fstream file;
     file.open("MEMBER.DAT",ios::out | ios::ate);
     strcpy(name,mname);
     strcpy(phone,mphone);
     strcpy(address,maddress);
     int location;
     location=(recno-1)*sizeof(MEMBER);
     file.seekp(location);
     file.write((char *)this,sizeof(MEMBER));
     file.close();
}
//add rec. in file for given mem. code
void MEMBER::addmem(int mcode,int bcode,char mname[26],char maddress[33],char
mphone[10],int d1,int m1,int y1)
{
     fstream file;
     file.open("MEMBER.DAT",ios::app);
     memcode=mcode;
     bookcode=bcode;
     strcpy(name,mname);
     strcpy(address,maddress);
     strcpy(phone,mphone);
     dd=d1;
     mm=m1;
     yy=y1;
     file.write((char *)this,sizeof(MEMBER));
     file.close();
}
//display rec. for given mem. code
void MEMBER::display(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               gotoxy(5,3);cout<<"Member Code #"<<mcode;
               gotoxy(5,4);cout<<"----------------------";
               gotoxy(5,6);cout<<"Name  :"<<name;
               gotoxy(5,7);cout<<"Phone :"<<phone;
               gotoxy(5,8);cout<<"Address:"<<address;
               break;
```

```
          }
     }
     file.close();
}
//display list of all members
void MEMBER::list(void)
{
     clrscr();
     BOOK B;
     int row=6,found=0,flag=0;
     char ch;
     gotoxy(32,2);cout<<"LIST OF MEMBERS";
     gotoxy(31,3);cout<<"----------------------------";
     gotoxy(1,4);cout<<"CODE   BOOK CODE    NAME                          PHONE";
     gotoxy(1,5);cout<<"------------------------------------------------------------
---";
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          flag=0;
          delay(20);
          found=1;
          gotoxy(2,row);cout<<memcode;
          gotoxy(10,row);cout<<bookcode;
          gotoxy(19,row);cout<<name;
          gotoxy(48,row);cout<<phone;
          textbackground(WHITE);textcolor(BLACK);
          gotoxy(7,row+1);
          if(bookcode==0)
          cprintf("BOOK NAME:(Not Issued)");
          else
          {
               cprintf("BOOK NAME:%s",B.bookname(bookcode));
               gotoxy(42,row+1);
               cprintf("Date of Return:");
               textcolor(BLACK+BLINK);
               cprintf("%d%d%d",dd,mm,yy);
          }
          textbackground(BLACK);textcolor(LIGHTGRAY);
          if(row==22)
          {
               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(32,2);cout<<"LIST OF MEMBERS";
               gotoxy(31,3);cout<<"--------------------";
               gotoxy(1,4);cout<<"CODE         BOOK CODE         NAME
PHONE";
               gotoxy(1,5);
               cout<<"------------------------------------------------------------
--------"
                   <<"------------------------------------------------------------
```

```
--------";
          }
          else
          row=row+2;
     }
     if(!found)
     {
          gotoxy(5,10);cout<<"\7 Records not found";
     }
     if(!flag)
     {
          gotoxy(1,25);cout<<"Press any key to continue...";
          getche();
     }
     file.close();
}
//GIVES DATA TO ADD RECORD IN BOOK FILE
void WORKING::addbook(void)
{
     if(!reccount())          //mem. func. of BOOK
     {
          addnewbook(0,"null","null",0.0,0,0);
          BOOK::deleterec(0);
     }
     char ch;
     int tcode,tcopies,tavail;
     char tname[33],tauthor[26];
     float tprice=0.0;
     do
     {
          int found=0,valid=0;
          int tc;
          float t2=0.0;
          char t[10],t1[10];
          clrscr();
          gotoxy(29,3);cout<<"ADDITION OF THE BOOKS";
          gotoxy(29,4);cout<<"----------------------";
          gotoxy(72,1);cout<<"<0>=Exit";
          gotoxy(5,25);cout<<"Enter code no. of the book";
          gotoxy(5,5);cout<<"Code no.";
          gets(t);
          tc=atoi(t);
          tcode=tc;
          if(tcode==0)return;
          if(bookfound(tcode))
          {
               found=1;
               gotoxy(19,8);cout<<bookname(tcode);
               gotoxy(19,9);cout<<authorname(tcode);
               gotoxy(22,10);cout<<bookprice(tcode);
          }
          gotoxy(5,8);cout<<"Book Name :";
          gotoxy(5,9);cout<<"Author Name:";
          gotoxy(5,10);cout<<"Price   :Rs.";
          gotoxy(5,12);cout<<"Copies   :";
          valid=0;
          while(!valid && !found)
          {
```

```
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the name of the book";
     gotoxy(19,8);clreol();
     gets(tname);
     strupr(tname);
     if(tname[0]=='0')return;
     if(strlen(tname)<1 || strlen(tname)>32)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly (Range:1..32)";
          getch();
     }
}
valid=0;
while(!valid && !found)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the author's name of the book";
     gotoxy(19,9);clreol();
     gets(tauthor);
     strupr(tauthor);
     if(tauthor[0]=='0')return;
     if(strlen(tauthor)<1 || strlen(tauthor)>25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly (Range:1..25)";
          getch();
     }
}
valid=0;
while(!valid && !found)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the price of the book";
     gotoxy(22,10);clreol();
     gets(t1);
     t2=atof(t1);
     tprice=t2;
     if(t1[0]=='0')return;
     if(tprice<1 || tprice > 9999)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly";
          getch();
     }
}
valid=0;
while(!valid)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter no. of copies of book to be added";
```

```
               gotoxy(19,12);clreol();
               gets(t);
               tc=atoi(t);
               tcopies=tc;
               if(t[0]=='0')return;
               if(tcopies<1 || tcopies > 50)
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7 Enter correctly";
                    getch();
               }
          }
          tavail=available(tcode) + tcopies;
          tcopies=noofcopies(tcode) + tcopies;
          gotoxy(5,25);clreol();
          do
          {
               gotoxy(5,15);clreol();
               cout<<"Do you want to save(y/n):";
               ch=getche();
               ch=toupper(ch);
          }while(ch != 'Y' && ch !='N');
          if(ch=='Y')
          {
               if(found)
               updatecopies(tcode,tcopies,tavail);
               else
               addnewbook(tcode,tname,tauthor,tprice,tcopies,tavail);
          }
          do
          {
               gotoxy(5,17);clreol();
               cout<<"Do you want to add more(y/n):";
               ch=getche();
               ch=toupper(ch);
          }while(ch != 'Y' && ch != 'N');
     }while(ch=='Y');
}
//gives data to add record in mem. file
void WORKING::addmember(void)
{
     char ch;
     int mcode,bcode;
     char mname[26],mphone[10],maddress[33];
     int d1,m1,y1;
     mcode=lastcode();
     mcode++;
     do
     {
          int valid=0;
          clrscr();
          gotoxy(28,3);
          cout<<"ADDITION OF THE MEMBERS";
          gotoxy(28,4);
          cout<<"--------------------------";
          gotoxy(72,1);
          cout<<"<0>=Exit";
```

```
          gotoxy(5,7);
          cout<<"Member Code #"<<mcode;
          gotoxy(5,8);
          cout<<"-----------------";
          gotoxy(5,10);cout<<"Name :";
          gotoxy(5,12);cout<<"Phone :";
          gotoxy(5,14);cout<<"Address:";
          do
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter the name of New Member";
               gotoxy(15,10);clreol();
               gets(mname);
               strupr(mname);
               if(mname[0]=='0')
               return;
               if(strlen(mname)<1 || strlen(mname) > 25)
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7Enter correctly(Range: 1..25)";
                    getch();
               }
          }while(!valid);
          do
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter Phone no. of the Member or Press <ENTER> for none";
               gotoxy(15,12);clreol();
               gets(mphone);
               if(mphone[0]=='0')return;
               if((strlen(mphone) < 7 && strlen(mphone) > 0) || (strlen(mphone) >
9))
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7Enter correctly";
                    getch();
               }
          }while(!valid);
          if(strlen(mphone)==0)strcpy(mphone,"-");
          do
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter the address of the New Member";
               gotoxy(15,14);clreol();
               gets(maddress);
               strupr(maddress);
               if(maddress[0]=='0')return;
               if(strlen(maddress)<1 || strlen(maddress)>32)
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7Enter correctly(Range: 1..32)";
                    getch();
```

```
               }
          }while(!valid);
          gotoxy(5,25);clreol();
          do
          {
               gotoxy(5,17);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=='Y')
          {
               bcode=0;
               d1=0;
               m1=0;y1=0;
               addmem(mcode,bcode,mname,maddress,mphone,d1,m1,y1);
               mcode++;
          }
          do
          {
               gotoxy(5,19);clreol();
               cout<<"Do you want to add more(y/n):";
               ch=getche();
               ch=toupper(ch);
               if(ch=='0')return;
          }while(ch!='Y' && ch!='N');
     }while(ch=='Y');
}
//issues the book
void WORKING::issuebook(void)
{
     BOOK B;
     MEMBER M;
     DATE D;
     char t1code[33],ch;
     int t2code=0,tcode=0,mcode=0;
     int valid;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be issued";
               gotoxy(5,6);cout<<"             or                  ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.........:";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
```

```
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
     tcode=bookcodeof(t1code);
     if(!available(tcode))
     {
          gotoxy(5,10);
          cout<<"\7Sorry! Book("<<bookname(tcode)<<") is not available";
          gotoxy(5,11);cout<<"Kindly issue another book";
          gotoxy(5,12);
          cout<<"See list of Books";
          getch();
          return;
     }
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(5,5);cout<<"Book Name:"<<bookname(tcode);
               gotoxy(5,7);cout<<"Enter Code no. of the Member";
               gotoxy(5,8);cout<<"             or                 ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.......: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               M.list();
               else
               break;
          }
          t2code=atoi(t1code);
          mcode=t2code;
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);
               cout<<"\7Enter Correctly";
               getch();
          }
          if(!memfound(mcode) && valid)
```

```
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
               gotoxy(5,14);cout<<"Press <ESC> to exit or any other Key to
continue...";
               ch=getch();
               if(ch==27)return;
          }
          if(issued(mcode) && valid)
          {
                valid=0;
                gotoxy(5,13);cout<<"\7Sorry!! you can not issue more than one";
                gotoxy(5,14);cout<<"Press <ESC> to exit or any other Key to
continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     int tcopies,tavail;
     tcopies=noofcopies(tcode);     //mem. func. of BOOK
     tavail=available(tcode)-1;
     updatecopies(tcode,tcopies,tavail);
     D.extenddate(d1,m1,y1,15);
     d1=D.day;
     m1=D.mon;
     y1=D.year;
     updatebook(mcode,tcode,d1,m1,y1);     //"      " " MEMBER
     gotoxy(5,13);
     cout<<"\7Book is issued to "<<memname(mcode);
     gotoxy(5,15);
     cout<<"Date of Return: "<<d1<<"/"<<m1<<"/"<<y1;
     getch();
}
void WORKING::returnbook(void)
{
     MEMBER M;
     char t1code[5],ch;
     int t2code=0,mcode=0;
     int valid;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(5,7);cout<<"Enter Code of the Member who wants to return
Book";
               gotoxy(5,8);cout<<"             or                  ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
               gets(t1code);
```

```
                if(t1code[0]=='0')return;
                if(strlen(t1code)==0)
                M.list();
                else
                break;
           }
           t2code=atoi(t1code);
           mcode=t2code;
           if(mcode==0)
           {
                valid=0;
                gotoxy(5,25);cout<<"\7Enter Correctly";
                getch();
           }
           if(!memfound(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Record not found";
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
           if(!issued(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Member have no book to return";
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
      }while(!valid);
      int bcode,tcopies,tavail;
      bcode=issued(mcode);
      gotoxy(5,13);cout<<"Book Code:"<<bcode;
      gotoxy(5,14);cout<<"Book Name:"<<bookname(bcode);
      tcopies=noofcopies(bcode);
      tavail=available(bcode)+1;
      int f;
      f=fine(mcode);
      if(f!=0)
      {
           gotoxy(5,16);cout<<"You have to pay a fine of Rs."<<f;
           gotoxy(5,17);cout<<"Please do not delay the Return of book again";
      }
      updatecopies(bcode,tcopies,tavail);
      updatebook(mcode,0,0,0,0);
      gotoxy(5,19);cout<<"\7Book has been returned";
      getch();
}
//gives data to modify book record
void WORKING::modifybook(void)
{
      BOOK B;
      char t1code[5],tname[33],tauthor[26],*t1,ch;
      int t2code=0,tcode=0;
      float t2=0.0,tprice=0.0;
```

```
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be modified";
               gotoxy(5,6);cout<<"             or                  ";
               gotoxy(5,7);cout<<"Press <ENTER> for help..........: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
     tcode=bookcodeof(t1code);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     BOOK::display(tcode);
     do
     {
          gotoxy(5,13);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;
     gotoxy(5,16);cout<<"Book Name :";
     gotoxy(5,17);cout<<"Author Name:";
     gotoxy(5,18);cout<<"Price   :Rs.";
     do
     {
          valid=1;
          gotoxy(5,25);clreol();
          cout<<"Enter the name of the book or <ENTER> for no change";
          gotoxy(19,16);clreol();
          gets(tname);
          strupr(tname);
          if(tname[0]=='0')return;
          if(strlen(tname)>32)
```

```
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter correctly(Range: 1..32)";
          getch();
      }
 }while(!valid);
 if(strlen(tname)==0)
 strcpy(tname,bookname(tcode));
 do
 {
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the author's name or <ENTER> for no change";
     gotoxy(19,17);clreol();
     gets(tauthor);
     strupr(tauthor);
     if(tauthor[0]=='0')return;
     if(strlen(tauthor)>25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter Correctly(Range: 1..25)";
          getch();
     }
 }while(!valid);
 if(strlen(tauthor)==0)
 strcpy(tauthor,authorname(tcode));
 do
 {
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter price or <ENTER> for no change";
     gotoxy(22,18);clreol();
     gets(t1);
     t2=atof(t1);
     tprice=t2;
     if(t1[0]=='0')return;
     if((tprice<1 || tprice > 9999) && (t1[0]!='\0'))
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly";
          getch();
     }
}while(!valid);
if(strlen(t1)==0)
tprice=bookprice(tcode);
gotoxy(5,25);
clreol();
do
{
     gotoxy(5,20);clreol();
     cout<<"Do you want to save changes(y/n)";
     ch=getche();
     ch=toupper(ch);
     if(ch=='0')return;
}while(ch!='Y' && ch!='N');
```

```
     if(ch=='N')
     return;
     BOOK::modify(tcode,tname,tauthor,tprice);
     gotoxy(5,23);cout<<"\7Record Modified";
     getch();
}
//gives data to modify member record
void WORKING::modifymem(void)
{
     MEMBER M;
     char m1code[10],mname[26],mphone[10],maddress[33],ch;
     int m2code=0,mcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,7);cout<<"Enter Code no. of the Member to be modified";
               gotoxy(5,8);cout<<"             or                 ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
               gets(m1code);
               m2code=atoi(m1code);
               mcode=m2code;
               if(m1code[0]=='0')return;
               if(strlen(m1code)==0)
               M.list();
               else
               break;
          }
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);cout<<"\7Enter Correctly";
               getch();
          }
          if(valid && !memfound(mcode))
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
               gotoxy(5,14);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     MEMBER::display(mcode);
     do
     {
          gotoxy(5,10);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;
gotoxy(5,13);cout<<"Name  :";
gotoxy(5,14);cout<<"Phone  :";
gotoxy(5,15);cout<<"Address:";
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the name of the member or <ENTER> for no change";
     gotoxy(19,13);clreol();
     gets(mname);
     strupr(mname);
     if(mname[0]=='0')return;
     if(strlen(mname) > 25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly(Range: 1..25)";
          getch();
     }
}while(!valid);
if(strlen(mname)==0)
strcpy(mname,memname(mcode));
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the Phone no. of Member or <ENTER> for no change";
     gotoxy(19,14);clreol();
     gets(mphone);
     if(mphone[0]=='0')return;
     if((strlen(mphone) < 7 && strlen(mphone) > 0 || (strlen(mphone) > 9)))
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter Correctly";
          getch();
     }
}while(!valid);
if(strlen(mphone)==0)
strcpy(mphone,memphone(mcode));
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the address of the member or <ENTER> for no change";
     gotoxy(19,15);clreol();
     gets(maddress);
     strupr(maddress);
     if(maddress[0]=='0')return;
     if(strlen(maddress) > 32)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly(Range: 1..32)";
          getch();
```

```
          }
     }while(!valid);
     if(strlen(maddress)==0)
     strcpy(maddress,memaddress(mcode));
     gotoxy(5,25);clreol();
     do
     {
          gotoxy(5,18);clreol();
          cout<<"Do you want to save changes(y/n)";
          ch=getche();
          ch=toupper(ch);
          if(ch=='0')return;
     }while(ch!='Y' && ch!='N');
     if(ch=='N')
     return;
     MEMBER::modify(mcode,mname,mphone,maddress);
     gotoxy(5,23);
     cout<<"\7Record Modified";
     getch();
}
//gives book code to delete book record
void WORKING::deletebook(void)
{
     BOOK B;
     char t1code[5],tname[33],tauthor[26],ch;
     int t2code=0,tcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be deleted";
               gotoxy(5,6);cout<<"             or                  ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.........: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
```

```
     tcode=bookcodeof(t1code);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     BOOK::display(tcode);
     do
     {
          gotoxy(5,13);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;
     int tavail,tcopies;
     tavail=available(tcode);
     tcopies=noofcopies(tcode);
     if(tavail!=tcopies)
     {
          gotoxy(5,15);cout<<"\7Record cannot be deleted.This book is issued.";
          gotoxy(5,16);cout<<"\7Type special code to delete";
          if(getch()!='7')
          {        gotoxy(5,19);cout<<"\7*";
                getch();
                return;
          }
          else
                gotoxy(5,19);cout<<"\7*";
     }
     BOOK::deleterec(tcode);
     gotoxy(5,23);cout<<"\7Record Deleted";
     getch();
}
//gives mem. code to delete member record
void WORKING::deletemem(void)
{
     MEMBER M;
     char m1code[10],mname[26],mphone[10],maddress[33],ch;
     int m2code=0,mcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,7);cout<<"Enter Code no. of the Member to be deleted";
               gotoxy(5,8);cout<<"             or                 ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.....: ";
               gets(m1code);
               m2code=atoi(m1code);
               mcode=m2code;
               if(m1code[0]=='0')return;
               if(strlen(m1code)==0)
               M.list();
               else
               break;
          }
```

```
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);cout<<"\7Enter Correctly";
               getch();
          }
          if(valid && !memfound(mcode))
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
               gotoxy(5,14);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     MEMBER::display(mcode);
     do
     {
          gotoxy(5,10);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;
     if(issued(mcode))
     {
          gotoxy(5,15);
          cout<<"\7Record cannot be deleted.Member has a book.";
          getch();
          return;
     }
     MEMBER::deleterec(mcode);
     gotoxy(5,23);cout<<"\7Record Modified";
     getch();
}
//Main func. calling introduction and main menu
void main()
{
     MENU menu;
     menu.introduc();
     menu.mainmenu();
}
```

```
#include<iostream.h>
#include<process.h>
#include<string.h>
#include<stdlib.h>
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<dos.h>
#include<fstream.h>
#include<graphics.h>
#include<math.h>
class MENU
{
     public:
          void mainmenu(void);
          void introduc(void);
          void bkgrnd(int);
          int bkgrnd(void);
          int check(int,int);
     private:
          void editmenu(void);
          void editbook(void);
          void editmem(void);
};
class BOOK
{
     public:
          void list(void);
          char *bookname(int);
     protected:
          void addnewbook(int,char tname[33],char tauthor[26],float,int,int);
          void updatecopies(int,int,int);
          void modify(int,char[],char[],float);
          void deletion(void);
          int bookfound(int);
          int booknamefound(char[]);
          int recordno(int);
          int available(int);
          char *authorname(int);
          float bookprice(int);
          int noofcopies(int);
          int bookcodeof(char[]);
          void display(int);
          int reccount(void);
          void deleterec(int);
     private:
          int bookcode,copies;
```

```
          char name[33],author[26];
          float price;
          int avail;
};
class MEMBER
{
     public:
          void list(void);
     protected:
          void addmem(int,int,char[],char[],char[],int,int,int);
          void modify(int,char[],char[],char[]);
          void deletion(void);
          int memfound(int);
          void updatebook(int,int,int,int,int);
          char *memname(int);
          char *memphone(int);
          char *memaddress(int);
          int recordno(int);
          int lastcode(void);
          int issued(int);
          int fine(int);
          void display(int);
          void deleterec(int);
     private:
          int memcode,bookcode;
          char name[26],phone[10],address[33];
          int dd,mm,yy;
};
class WORKING:public BOOK,public MEMBER
{
     public:
          void issuebook(void);
          void returnbook(void);
          void addbook(void);
          void addmember(void);
          void modifybook(void);
          void modifymem(void);
          void deletebook(void);
          void deletemem(void);
};
class DATE
{
     public:
          void extenddate(int,int,int,int);
          int diff(int,int,int,int,int,int);
          int day,mon,year;
};
void DATE::extenddate(int d1,int m1,int y1,int days)
{
     static int month[]={31,29,31,30,31,30,31,31,30,31,30,31};
     for(int i=1;i<=days;i++)
     {
          d1++;
          if((d1>month[m1-1]) || (y1%4!=0 && m1==2 && d1>28))
          {
               d1=1;
               m1++;
          }
```

```
          if(m1>12)
          {
               m1=1;
               y1++;
          }
     }
     day=d1;
     mon=m1;
     year=y1;
}
int DATE::diff(int d1,int m1,int y1,int d2,int m2,int y2)
{
     int days=0;
     if((y2<y1) || (y2==y1 && m2<m1) || (y2==y1 && m2==m1 && d2<d1))
     return days;
     static int month[]={31,29,31,30,31,30,31,31,30,31,30,31};
     while(d1!=d2 || m1!=m2 || y1!=y2)
     {
          days++;
          d1++;
          if((d1>month[m1-1]) || (y1%4!=0 && m1==2 && d1>28))
          {
               d1=1;
               m1++;
          };
          if(m1>12)
          {
               m1=1;
               y1++;
          }
     }
     return days;
}
void MENU::mainmenu(void)
{

     char ch;
     int p;
     MENU::bkgrnd(0);
     while(1)
     {
           // cleardevice();


          gotoxy(29,6);
          cout<<"BOOK LIBRARY";
          gotoxy(29,7);
          cout<<"-------------------------------";
          gotoxy(30,10);cout<<"1.Introduction";
          gotoxy(30,11);cout<<"2.Add New Book(s)";
          gotoxy(30,12);cout<<"3.Add New Member(s)";
          gotoxy(30,13);cout<<"4.Issue Book(s)";
          gotoxy(30,14);cout<<"5.Return Book(s)";
          gotoxy(30,15);cout<<"6.List of Book(s)";
          gotoxy(30,16);cout<<"7.List of Member(s)";
          gotoxy(30,17);cout<<"8.EDIT";
          gotoxy(30,18);cout<<"0.QUIT";
          gotoxy(30,20);cout<<"Enter Your Choice:";
```

```
          ch=getch();
          closegraph();
          if(ch==27)
          break;
          if(ch=='1')
          introduc();
          else
          if(ch=='2')
          {
               WORKING W;
               W.addbook();
          }
          else
          if(ch=='3')
          {
               WORKING W;
               W.addmember();
          }
          else
          if(ch=='4')
          {
               WORKING W;
               W.issuebook();
          }
          else
          if(ch=='5')
          {
               WORKING W;
               W.returnbook();
          }
          else
          if(ch=='6')
          {
               BOOK B;
               B.list();
          }
          else
          if(ch=='7')
          {
               MEMBER M;
               M.list();
          }
          else
          if(ch=='8')
          editmenu();
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//function for bkground
void MENU::bkgrnd(int v)
 {
     clrscr();
    int   driver = DETECT,mode;
    int   x[15],y[15];
    int   x_center , y_center , rad = 90;
```

```
   int   i,j;

   initgraph(&driver,&mode,"c:\\tc\\bgi");
    x_center=getmaxx()/2;
    y_center=getmaxy()/2;

    setbkcolor(0);
    setcolor(BLUE);
    if(v==0)
      rad=100;
    if(v==1)
      rad=2000;
    for (;rad<=2000;rad+=50)
    {
        for ( i = 0; i < 15; i++ )
         {
             x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
             y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
         }
        for ( i = 0; i < 15; i++ )
         for ( j = 0; j < 15; j++ )
           {
              line(x[i],y[i],x[j],y[j]);
                if(v==1)
              delay(20);
           }

        if(rad==2000)
          break;

        cleardevice();
    }
}

int MENU::bkgrnd(void)
{
    clrscr();
   int   driver = DETECT,mode;
   int   x[15],y[15];
   int   x_center , y_center , rad = 2000;
   int   i,j,k;
   int   color=0,maxcolor,v=1;


   initgraph(&driver,&mode,"c:\\tc\\bgi");
    x_center=getmaxx()/2;
    y_center=getmaxy()/2;
    maxcolor=getmaxcolor();

    setbkcolor(0);
    while(k=(check(0,7))==9)
    {    setcolor(color);
        for ( i = 0; i < 15; i++ )
         {
             x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
             y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
         }
        for ( i = 0; i < 15; i++ )
```

```
          for ( j = 0; j < 15; j++ )
                 line(x[i],y[i],x[j],y[j]);


         if(color==maxcolor)
            v=-1;
         if(color==0)
            v=1;

         delay(500);
         color+=v;
        // cleardevice();
     }
     return(k);
 }
int MENU::check(int n,int m)
{int i,flag,temp;
char ch[10];
gets(ch);
temp=atoi(ch);
for(i=n;i<=m;i++)
if(temp==i)
{flag=1;
break;}

if(flag==1)
return(temp);
else
return(9);
}
//function display edit menu
void MENU::editmenu(void)
     {
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          //clrscr();

          gotoxy(32,9);cout<<"EDIT MENU";
          gotoxy(32,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.BOOKS";
          gotoxy(34,14);cout<<"2.MEMBERS";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(31,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          editbook();
          else
          if(ch=='2')
          editmem();
          else
          if(ch=='0')
          break;
```

```
          MENU::bkgrnd(1);
     }
}
//func. to display edit menu for book and control
//all the func. in edit menu
void MENU::editbook(void)
{
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          //clrscr();
          gotoxy(31,9);cout<<"EDIT BOOKS";
          gotoxy(31,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.MODIFY";
          gotoxy(34,14);cout<<"2.DELETE";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(31,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
          break;
          else
          if(ch=='1')
          {
               WORKING W;
               W.modifybook();
          }
          else
          if(ch=='2')
          {
               WORKING W;
               W.deletebook();
          }
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display edit menu for member and control
//all the func. in edit menu
void MENU::editmem(void)
{
     char ch;
     MENU::bkgrnd(0);
     while(1)
     {
          clrscr();
          gotoxy(29,9);cout<<"EDIT MEMBERS";
          gotoxy(29,10);cout<<"--------------------------";
          gotoxy(34,13);cout<<"1.MODIFY";
          gotoxy(34,14);cout<<"2.DELETE";
          gotoxy(34,15);cout<<"0.EXIT";
          gotoxy(29,17);cout<<"Enter Your choice:";
          ch=getche();
          closegraph();
          if(ch==27)
```

```
          break;
          else
          if(ch=='1')
          {
               WORKING W;
               W.modifymem();
          }
          else
          if(ch=='2')
          {
               WORKING W;
               W.deletemem();
          }
          else
          if(ch=='0')
          break;
          MENU::bkgrnd(1);
     }
}
//func. to display the introduction for the project
void MENU::introduc(void)
{
    clrscr();

    int   driver = DETECT,mode;
    int   x[15],y[15];
    int   x_center = 360, y_center = 180, rad = 90;
    int   i,j;
    char  *fname[]={"P","R" ,"O" ,"J","E","C","T","L","I","B","R","A","R","Y"};
    char  *fname2[]={"M","A" ,"D" ,"E "," ","B","Y","
","S","I","D","H","A","R","T","H","&"," ","N","I","K","H","I","L"," "};

    initgraph(&driver,&mode,"c:\\tc\\bgi");

    setbkcolor(0);
    setcolor(BLUE);
    delay(1000);
     for (x_center=110;x_center<=520;x_center+=10)
     {    for ( i = 0; i < 15; i++ )
          {
                x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
                y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
          }
         for ( i = 0; i < 15; i++ )
               for ( j = 0; j < 15; j++ )
                    {
                         line(x[i],y[i],x[j],y[j]);
                    }
               delay(100);
               if(x_center!=520)
               cleardevice();
     }
     setcolor(WHITE);
     settextstyle(4, HORIZ_DIR, 5);


     for(i=0;i<14;i++)
      {
```

```
        outtextxy(40+40*i,120+50*(i/7), *(fname+i));  /* output a message */
        delay(300);
     }
    setcolor(RED);
    settextstyle(0, HORIZ_DIR, 1);

    for(i=0;i<25;i++)
     {
        outtextxy(440+10*i-70*(i/16),440+12*(i/8), *(fname2+i));
        delay(100);
     }
    setcolor(BLUE);
    int  midx=getmaxx()/2;
      y_center=getmaxy()/2;
    for (x_center=520;x_center>=midx;x_center-=10)
      {    for ( i = 0; i < 15; i++ )
          {
              x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
              y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
          }
           for ( i = 0; i < 15; i++ )
             for ( j = 0; j < 15; j++ )
                {
                 line(x[i],y[i],x[j],y[j]);
                }
           delay(100);
           if(x_center!=midx)
           cleardevice();
      }
   for (rad=100;rad<=2500;rad+=50)
     {
           for ( i = 0; i < 15; i++ )
           {
               x[i] =  x_center + rad *  cos((24*i+(x_center-110))*3.14159/180);
               y[i] =  y_center + rad *  sin((24*i+(x_center-110))*3.14159/180);
           }
           for ( i = 0; i < 15; i++ )
           for ( j = 0; j < 15; j++ )
              {
              line(x[i],y[i],x[j],y[j]);
              }
           delay(100);
           if(rad==2000)
          break;
           cleardevice();
     }
  // getch();           /* press any key return to TEXT mode */
  // closegraph();
    gotoxy(31,5);cout<<"Welcome To Project";
    textcolor(RED+BLINK);textbackground(WHITE);
    gotoxy(33,7);cprintf("BOOK LIBRARY");
    textcolor(LIGHTGRAY);textbackground(BLACK);
    gotoxy(15,10);cout<<"This project has facility "<<"of maintaining records";
    gotoxy(15,11);cout<<"of BOOKS & MEMBERS.";
    gotoxy(15,13);cout<<"This project can hold "<<"more than 10,000 books";
    gotoxy(15,14);cout<<"records.";
    gotoxy(15,16);cout<<"One member can issue one"<<" book at a time.If he/she";
    gotoxy(15,17);cout<<"does not return book upto "<<"15 days he/she have to ";
```

```
     gotoxy(15,18);cout<<"pay fine of Rs.2/- per day.";
     textcolor(LIGHTGRAY+BLINK);
     gotoxy(27,24);cprintf("Press any key to continue");
     textcolor(LIGHTGRAY);
     getch();
     closegraph();
}
//this func. return 0 if given book code not found
int BOOK::bookfound(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//return 0 if book name not found
int BOOK::booknamefound(char t1code[33])
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(!strcmpi(name,t1code))
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//return record no. for book code
int BOOK::recordno(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          count++;
          if(bookcode==tcode)break;
     }
     file.close();
     return count;
}
//returns available copies for given book code
int BOOK::available(int tcode)
```

```
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tavail=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tavail=avail;break;
          }
     }
     file.close();
     return tavail;
}
//return no. of copies for given book code
int BOOK::noofcopies(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tcopies=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tcopies=copies;break;
          }
     }
     file.close();
     return tcopies;
}
//return book name of the given book code
char *BOOK::bookname(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     char tname[33];
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               strcpy(tname,name);break;
          }
     }
     file.close();
     return tname;
}
//returns author name of the given book code
char *BOOK::authorname(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     char tauthor[26];
     while(file.read((char *)this,sizeof(BOOK)))
     {
```

```
          if(bookcode==tcode)
          {
               strcpy(tauthor,author);break;
          }
     }
     file.close();
     return tauthor;
}
//return book price of the given book code
float BOOK::bookprice(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     float tprice=0.0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               tprice=price;break;
          }
     }
     file.close();
     return tprice;
}
//return book code of the given book name
int BOOK::bookcodeof(char t1code[33])
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int tcode=0;
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(!strcmpi(name,t1code))
          {
               tcode=bookcode;break;
          }
     }
     file.close();
     return tcode;
}
//return no. of records in the book file
int BOOK::reccount(void)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(BOOK)))
     count++;
     file.close();
     return count;
}
//deletes record of the given book code
void BOOK::deleterec(int tcode)
{
     fstream file;
```

```
     file.open("BOOK.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(BOOK));
          if(file.eof())break;
          if(bookcode!=tcode)
          temp.write((char *)this,sizeof(BOOK));
     }
     file.close();
     temp.close();
     file.open("BOOK.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(BOOK));
          if(temp.eof())break;
          if(bookcode!=tcode)
          file.write((char *)this,sizeof(BOOK));
     }
     file.close();
     temp.close();
}
//add record in book file
void BOOK::addnewbook(int tcode,char tname[33],char tauthor[26],float tprice,int
tcopies,int tavail)
{
     fstream file;
     file.open("BOOK.DAT",ios::app);
     bookcode=tcode;
     strcpy(name,tname);
     strcpy(author,tauthor);
     price=tprice;
     copies=tcopies;
     avail=tavail;
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//updates the record in book file for given code
void BOOK::updatecopies(int tcode,int tcopies,int tavail)
{
     int recno;
     recno=recordno(tcode);
     fstream file;
     file.open("BOOK.DAT",ios::out | ios::ate);
     copies=tcopies;
     avail=tavail;
     int location;
     location=(recno-1)*sizeof(BOOK);
     file.seekp(location);
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//modify record in book file for given code
void BOOK::modify(int tcode,char tname[33],char tauthor[26],float tprice)
```

```
{
     int recno;
     recno=recordno(tcode);
     fstream file;
     file.open("BOOK.DAT",ios::out | ios::ate);
     strcpy(name,tname);
     strcpy(author,tauthor);
     price=tprice;
     int location;
     location=(recno-1)*sizeof(BOOK);
     file.seekp(location);
     file.write((char *)this,sizeof(BOOK));
     file.close();
}
//display record from book file for given book code
void BOOK::display(int tcode)
{
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(BOOK)))
     {
          if(bookcode==tcode)
          {
               gotoxy(5,5);cout<<"Book Code :"<<bookcode;
               gotoxy(5,7);cout<<"Book Name :"<<name;
               gotoxy(5,8);cout<<"Author Name :"<<author;
               gotoxy(5,9);cout<<"Price :Rs. "<<price;
               gotoxy(5,10);cout<<"Copies   :"<<copies;
               gotoxy(5,11);cout<<"Available  :"<<avail;
               break;
          }
     }
     file.close();
}
//display list of books
void BOOK::list(void)
{
     clrscr();
     int row=6,found=0,flag=0;
     char ch;
     gotoxy(33,2);cout<<"  LIST OF BOOKS";
     gotoxy(32,3);cout<<"-------------------";
     gotoxy(1,4);cout<<"CODE      BOOK NAME                    AUTHOR
PRICE  COPIES";
     gotoxy(1,5);cout<<"------------------------------------------------------------
---------------------";
     fstream file;
     file.open("BOOK.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(BOOK)))
     {
          flag=0;
          delay(20);
          found=1;
          gotoxy(2,row);cout<<bookcode;
          gotoxy(10,row);cout<<name;
          gotoxy(41,row);cout<<author;
```

```
          gotoxy(64,row);cout<<price;
          gotoxy(71,row);cout<<copies;
          textbackground(WHITE);textcolor(BLACK);
          gotoxy(40,row+1);cprintf("STATUS:");
          textcolor(BLACK+BLINK);
          cprintf("%d copies available",avail);
          textbackground(BLACK);textcolor(LIGHTGRAY);
          if(row==22)
          {
               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(33,2);cout<<"LIST OF BOOKS";
               gotoxy(32,3);cout<<"--------------------";
               gotoxy(1,4);cout<<"CODE        BOOK NAME          "
                         <<"         AUTHOR      PRICE  COPIES";
               gotoxy(1,5);
               cout<<"-------------------------------------------------"
                   <<"-------------------------------------------------";
          }
          else
          row=row+2;
     }
     if(!found)
     {
          gotoxy(5,10);cout<<"\t Records not found";
     }
     if(!flag)
     {
          gotoxy(1,25);cout<<"Press any key to continue...";
          getche();
     }
     file.close();
}
//returns 0 if given member code not found
int MEMBER::memfound(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int found=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               found=1;break;
          }
     }
     file.close();
     return found;
}
//returns 0 if member have not issued any book
int MEMBER::issued(int mcode)
{
```

```
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int missue=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               missue=bookcode;break;
          }
     }
     file.close();
     return missue;
}
//calculate & return fine for given mem. code
int MEMBER::fine(int mcode)
{
     DATE D;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int days,t_fine;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               days=D.diff(dd,mm,yy,d1,m1,y1);
               t_fine=days*2;
               break;
          }
     }
     file.close();
     return t_fine;
}
int MEMBER::lastcode(void)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int mcode=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     memcode=mcode;
     file.close();
     return mcode;
}
//returns mem. name given membercode
char *MEMBER::memname(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char mname[26];
```

```
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(mname,name);break;
          }
     }
     file.close();
     return mname;
}
//returns mem. add. of given mem. code
char *MEMBER::memphone(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char mphone[10];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(mphone,phone);break;
          }
     }
     file.close();
     return mphone;
}
char *MEMBER::memaddress(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     char maddress[33];
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               strcpy(maddress,address);break;
          }
     }
     file.close();
     return maddress;
}
//returns rec. no. of given mem.
int MEMBER::recordno(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     int count=0;
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          count++;
          if(memcode==mcode)break;

     }
     file.close();
     return count;
```

```
}

//returns record no. of given member code
void MEMBER::deleterec(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(MEMBER));
          if(file.eof())break;
          if(memcode!=mcode)
          temp.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
     file.open("MEMBER.DAT",ios::out);
     temp.open("temp.dat",ios::in);
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(MEMBER));
          if(temp.eof())break;
          file.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
}
//update record for given mem. code
void MEMBER::updatebook(int mcode,int tcode,int d1,int m1,int y1)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     fstream temp;
     temp.open("temp.dat",ios::out);
     file.seekg(0,ios::beg);
     while(!file.eof())
     {
          file.read((char *)this,sizeof(MEMBER));
          if(file.eof())break;
          if(memcode==mcode)
          {
               bookcode=tcode;
               dd=d1;
               mm=m1;
               yy=y1;
               temp.write((char *)this,sizeof(MEMBER));
          }
          else
          temp.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
     file.open("MEMBER.DAT",ios::out);
     temp.open("temp.dat",ios::in);
```

```
     temp.seekg(0,ios::beg);
     while(!temp.eof())
     {
          temp.read((char *)this,sizeof(MEMBER));
          if(temp.eof())break;
          file.write((char *)this,sizeof(MEMBER));
     }
     file.close();
     temp.close();
}
//modify record for given mem. code
void MEMBER::modify(int mcode,char mname[26],char mphone[10],char maddress[33])
{
     int recno;
     recno=recordno(mcode);
     fstream file;
     file.open("MEMBER.DAT",ios::out | ios::ate);
     strcpy(name,mname);
     strcpy(phone,mphone);
     strcpy(address,maddress);
     int location;
     location=(recno-1)*sizeof(MEMBER);
     file.seekp(location);
     file.write((char *)this,sizeof(MEMBER));
     file.close();
}
//add rec. in file for given mem. code
void MEMBER::addmem(int mcode,int bcode,char mname[26],char maddress[33],char
mphone[10],int d1,int m1,int y1)
{
     fstream file;
     file.open("MEMBER.DAT",ios::app);
     memcode=mcode;
     bookcode=bcode;
     strcpy(name,mname);
     strcpy(address,maddress);
     strcpy(phone,mphone);
     dd=d1;
     mm=m1;
     yy=y1;
     file.write((char *)this,sizeof(MEMBER));
     file.close();
}
//display rec. for given mem. code
void MEMBER::display(int mcode)
{
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          if(memcode==mcode)
          {
               gotoxy(5,3);cout<<"Member Code #"<<mcode;
               gotoxy(5,4);cout<<"----------------------";
               gotoxy(5,6);cout<<"Name  :"<<name;
               gotoxy(5,7);cout<<"Phone :"<<phone;
               gotoxy(5,8);cout<<"Address:"<<address;
```

```
               break;
          }
     }
     file.close();
}
//display list of all members
void MEMBER::list(void)
{
     clrscr();
     BOOK B;
     int row=6,found=0,flag=0;
     char ch;
     gotoxy(32,2);cout<<"LIST OF MEMBERS";
     gotoxy(31,3);cout<<"----------------------------";
     gotoxy(1,4);cout<<"CODE   BOOK CODE    NAME                          PHONE";
     gotoxy(1,5);cout<<"------------------------------------------------------------
---";
     fstream file;
     file.open("MEMBER.DAT",ios::in);
     file.seekg(0,ios::beg);
     while(file.read((char *)this,sizeof(MEMBER)))
     {
          flag=0;
          delay(20);
          found=1;
          gotoxy(2,row);cout<<memcode;
          gotoxy(10,row);cout<<bookcode;
          gotoxy(19,row);cout<<name;
          gotoxy(48,row);cout<<phone;
          textbackground(WHITE);textcolor(BLACK);
          gotoxy(7,row+1);
          if(bookcode==0)
          cprintf("BOOK NAME:(Not Issued)");
          else
          {
               cprintf("BOOK NAME:%s",B.bookname(bookcode));
               gotoxy(42,row+1);
               cprintf("Date of Return:");
               textcolor(BLACK+BLINK);
               cprintf("%d%d%d",dd,mm,yy);
          }
          textbackground(BLACK);textcolor(LIGHTGRAY);
          if(row==22)
          {
               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(32,2);cout<<"LIST OF MEMBERS";
               gotoxy(31,3);cout<<"--------------------";
               gotoxy(1,4);cout<<"CODE         BOOK CODE         NAME
PHONE";
               gotoxy(1,5);
               cout<<"------------------------------------------------------------
--------"
```

```
                    <<"-------------------------------------------------------------
--------";
          }
          else
          row=row+2;
     }
     if(!found)
     {
          gotoxy(5,10);cout<<"\7 Records not found";
     }
     if(!flag)
     {
          gotoxy(1,25);cout<<"Press any key to continue...";
          getche();
     }
     file.close();
}
//GIVES DATA TO ADD RECORD IN BOOK FILE
void WORKING::addbook(void)
{
     if(!reccount())          //mem. func. of BOOK
     {
          addnewbook(0,"null","null",0.0,0,0);
          BOOK::deleterec(0);
     }
     char ch;
     int tcode,tcopies,tavail;
     char tname[33],tauthor[26];
     float tprice=0.0;
     do
     {
          int found=0,valid=0;
          int tc;
          float t2=0.0;
          char t[10],t1[10];
          clrscr();
          gotoxy(29,3);cout<<"ADDITION OF THE BOOKS";
          gotoxy(29,4);cout<<"----------------------";
          gotoxy(72,1);cout<<"<0>=Exit";
          gotoxy(5,25);cout<<"Enter code no. of the book";
          gotoxy(5,5);cout<<"Code no.";
          gets(t);
          tc=atoi(t);
          tcode=tc;
          if(tcode==0)return;
          if(bookfound(tcode))
          {
               found=1;
               gotoxy(19,8);cout<<bookname(tcode);
               gotoxy(19,9);cout<<authorname(tcode);
               gotoxy(22,10);cout<<bookprice(tcode);
          }
          gotoxy(5,8);cout<<"Book Name :";
          gotoxy(5,9);cout<<"Author Name:";
          gotoxy(5,10);cout<<"Price   :Rs.";
          gotoxy(5,12);cout<<"Copies   :";
          valid=0;
          while(!valid && !found)
```

```
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the name of the book";
     gotoxy(19,8);clreol();
     gets(tname);
     strupr(tname);
     if(tname[0]=='0')return;
     if(strlen(tname)<1 || strlen(tname)>32)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly (Range:1..32)";
          getch();
     }
}
valid=0;
while(!valid && !found)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the author's name of the book";
     gotoxy(19,9);clreol();
     gets(tauthor);
     strupr(tauthor);
     if(tauthor[0]=='0')return;
     if(strlen(tauthor)<1 || strlen(tauthor)>25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly (Range:1..25)";
          getch();
     }
}
valid=0;
while(!valid && !found)
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the price of the book";
     gotoxy(22,10);clreol();
     gets(t1);
     t2=atof(t1);
     tprice=t2;
     if(t1[0]=='0')return;
     if(tprice<1 || tprice > 9999)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly";
          getch();
     }
}
valid=0;
while(!valid)
{
     valid=1;
     gotoxy(5,25);clreol();
```

```
                cout<<"Enter no. of copies of book to be added";
                gotoxy(19,12);clreol();
                gets(t);
                tc=atoi(t);
                tcopies=tc;
                if(t[0]=='0')return;
                if(tcopies<1 || tcopies > 50)
                {
                     valid=0;
                     gotoxy(5,25);clreol();
                     cout<<"\7 Enter correctly";
                     getch();
                }
           }
           tavail=available(tcode) + tcopies;
           tcopies=noofcopies(tcode) + tcopies;
           gotoxy(5,25);clreol();
           do
           {
                gotoxy(5,15);clreol();
                cout<<"Do you want to save(y/n):";
                ch=getche();
                ch=toupper(ch);
           }while(ch != 'Y' && ch !='N');
           if(ch=='Y')
           {
                if(found)
                updatecopies(tcode,tcopies,tavail);
                else
                addnewbook(tcode,tname,tauthor,tprice,tcopies,tavail);
           }
           do
           {
                gotoxy(5,17);clreol();
                cout<<"Do you want to add more(y/n):";
                ch=getche();
                ch=toupper(ch);
           }while(ch != 'Y' && ch != 'N');
     }while(ch=='Y');
}
//gives data to add record in mem. file
void WORKING::addmember(void)
{
     char ch;
     int mcode,bcode;
     char mname[26],mphone[10],maddress[33];
     int d1,m1,y1;
     mcode=lastcode();
     mcode++;
     do
     {
          int valid=0;
          clrscr();
          gotoxy(28,3);
          cout<<"ADDITION OF THE MEMBERS";
          gotoxy(28,4);
          cout<<"--------------------------";
          gotoxy(72,1);
```

```
          cout<<"<0>=Exit";
          gotoxy(5,7);
          cout<<"Member Code #"<<mcode;
          gotoxy(5,8);
          cout<<"-----------------";
          gotoxy(5,10);cout<<"Name :";
          gotoxy(5,12);cout<<"Phone :";
          gotoxy(5,14);cout<<"Address:";
          do
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter the name of New Member";
               gotoxy(15,10);clreol();
               gets(mname);
               strupr(mname);
               if(mname[0]=='0')
               return;
               if(strlen(mname)<1 || strlen(mname) > 25)
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7Enter correctly(Range: 1..25)";
                    getch();
               }
          }while(!valid);
          do
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter Phone no. of the Member or Press <ENTER> for none";
               gotoxy(15,12);clreol();
               gets(mphone);
               if(mphone[0]=='0')return;
               if((strlen(mphone) < 7 && strlen(mphone) > 0) || (strlen(mphone) >
9))
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7Enter correctly";
                    getch();
               }
          }while(!valid);
          if(strlen(mphone)==0)strcpy(mphone,"-");
          do
          {
               valid=1;
               gotoxy(5,25);clreol();
               cout<<"Enter the address of the New Member";
               gotoxy(15,14);clreol();
               gets(maddress);
               strupr(maddress);
               if(maddress[0]=='0')return;
               if(strlen(maddress)<1 || strlen(maddress)>32)
               {
                    valid=0;
                    gotoxy(5,25);clreol();
                    cout<<"\7Enter correctly(Range: 1..32)";
```

```
                    getch();
               }
          }while(!valid);
          gotoxy(5,25);clreol();
          do
          {
               gotoxy(5,17);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=='Y')
          {
               bcode=0;
               d1=0;
               m1=0;y1=0;
               addmem(mcode,bcode,mname,maddress,mphone,d1,m1,y1);
               mcode++;
          }
          do
          {
               gotoxy(5,19);clreol();
               cout<<"Do you want to add more(y/n):";
               ch=getche();
               ch=toupper(ch);
               if(ch=='0')return;
          }while(ch!='Y' && ch!='N');
     }while(ch=='Y');
}
//issues the book
void WORKING::issuebook(void)
{
     BOOK B;
     MEMBER M;
     DATE D;
     char t1code[33],ch;
     int t2code=0,tcode=0,mcode=0;
     int valid;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be issued";
               gotoxy(5,6);cout<<"             or                  ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.........:";
               gets(t1code);
               if(t1code[0]=='0')return;
```

```
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
     tcode=bookcodeof(t1code);
     if(!available(tcode))
     {
          gotoxy(5,10);
          cout<<"\7Sorry! Book("<<bookname(tcode)<<") is not available";
          gotoxy(5,11);cout<<"Kindly issue another book";
          gotoxy(5,12);
          cout<<"See list of Books";
          getch();
          return;
     }
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(5,5);cout<<"Book Name:"<<bookname(tcode);
               gotoxy(5,7);cout<<"Enter Code no. of the Member";
               gotoxy(5,8);cout<<"             or                 ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.......: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               M.list();
               else
               break;
          }
          t2code=atoi(t1code);
          mcode=t2code;
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);
               cout<<"\7Enter Correctly";
               getch();
          }
```

```
          if(!memfound(mcode) && valid)
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
               gotoxy(5,14);cout<<"Press <ESC> to exit or any other Key to
continue...";
               ch=getch();
               if(ch==27)return;
          }
          if(issued(mcode) && valid)
          {
                valid=0;
                gotoxy(5,13);cout<<"\7Sorry!! you can not issue more than one";
                gotoxy(5,14);cout<<"Press <ESC> to exit or any other Key to
continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     int tcopies,tavail;
     tcopies=noofcopies(tcode);     //mem. func. of BOOK
     tavail=available(tcode)-1;
     updatecopies(tcode,tcopies,tavail);
     D.extenddate(d1,m1,y1,15);
     d1=D.day;
     m1=D.mon;
     y1=D.year;
     updatebook(mcode,tcode,d1,m1,y1);     //"      " " MEMBER
     gotoxy(5,13);
     cout<<"\7Book is issued to "<<memname(mcode);
     gotoxy(5,15);
     cout<<"Date of Return: "<<d1<<"/"<<m1<<"/"<<y1;
     getch();
}
void WORKING::returnbook(void)
{
     MEMBER M;
     char t1code[5],ch;
     int t2code=0,mcode=0;
     int valid;
     int d1,m1,y1;
     struct date d;
     getdate(&d);
     d1=d.da_day;
     m1=d.da_mon;
     y1=d.da_year;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,2);cout<<"Date:"<<d1<<"/"<<m1<<"/"<<y1;
               gotoxy(5,7);cout<<"Enter Code of the Member who wants to return
Book";
               gotoxy(5,8);cout<<"             or                  ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
```

```
                gets(t1code);
                if(t1code[0]=='0')return;
                if(strlen(t1code)==0)
                M.list();
                else
                break;
           }
           t2code=atoi(t1code);
           mcode=t2code;
           if(mcode==0)
           {
                valid=0;
                gotoxy(5,25);cout<<"\7Enter Correctly";
                getch();
           }
           if(!memfound(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Record not found";
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
           if(!issued(mcode) && valid)
           {
                valid=0;
                gotoxy(5,13);cout<<"\7Member have no book to return";
                gotoxy(5,14);
                cout<<"Press <ESC> to exit or any other key to continue...";
                ch=getch();
                if(ch==27)return;
           }
      }while(!valid);
      int bcode,tcopies,tavail;
      bcode=issued(mcode);
      gotoxy(5,13);cout<<"Book Code:"<<bcode;
      gotoxy(5,14);cout<<"Book Name:"<<bookname(bcode);
      tcopies=noofcopies(bcode);
      tavail=available(bcode)+1;
      int f;
      f=fine(mcode);
      if(f!=0)
      {
           gotoxy(5,16);cout<<"You have to pay a fine of Rs."<<f;
           gotoxy(5,17);cout<<"Please do not delay the Return of book again";
      }
      updatecopies(bcode,tcopies,tavail);
      updatebook(mcode,0,0,0,0);
      gotoxy(5,19);cout<<"\7Book has been returned";
      getch();
}
//gives data to modify book record
void WORKING::modifybook(void)
{
      BOOK B;
      char t1code[5],tname[33],tauthor[26],*t1,ch;
      int t2code=0,tcode=0;
```

```
     float t2=0.0,tprice=0.0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be modified";
               gotoxy(5,6);cout<<"             or                 ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.....: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     if(tcode==0)
     tcode=bookcodeof(t1code);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     BOOK::display(tcode);
     do
     {
          gotoxy(5,13);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;
     gotoxy(5,16);cout<<"Book Name :";
     gotoxy(5,17);cout<<"Author Name:";
     gotoxy(5,18);cout<<"Price   :Rs.";
     do
     {
          valid=1;
          gotoxy(5,25);clreol();
          cout<<"Enter the name of the book or <ENTER> for no change";
          gotoxy(19,16);clreol();
          gets(tname);
          strupr(tname);
          if(tname[0]=='0')return;
```

```
     if(strlen(tname)>32)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter correctly(Range: 1..32)";
          getch();
      }
 }while(!valid);
 if(strlen(tname)==0)
 strcpy(tname,bookname(tcode));
 do
 {
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the author's name or <ENTER> for no change";
     gotoxy(19,17);clreol();
     gets(tauthor);
     strupr(tauthor);
     if(tauthor[0]=='0')return;
     if(strlen(tauthor)>25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter Correctly(Range: 1..25)";
          getch();
     }
 }while(!valid);
 if(strlen(tauthor)==0)
 strcpy(tauthor,authorname(tcode));
 do
 {
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter price or <ENTER> for no change";
     gotoxy(22,18);clreol();
     gets(t1);
     t2=atof(t1);
     tprice=t2;
     if(t1[0]=='0')return;
     if((tprice<1 || tprice > 9999) && (t1[0]!='\0'))
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly";
          getch();
     }
}while(!valid);
if(strlen(t1)==0)
tprice=bookprice(tcode);
gotoxy(5,25);
clreol();
do
{
     gotoxy(5,20);clreol();
     cout<<"Do you want to save changes(y/n)";
     ch=getche();
     ch=toupper(ch);
     if(ch=='0')return;
```

```
     }while(ch!='Y' && ch!='N');
     if(ch=='N')
     return;
     BOOK::modify(tcode,tname,tauthor,tprice);
     gotoxy(5,23);cout<<"\7Record Modified";
     getch();
}
//gives data to modify member record
void WORKING::modifymem(void)
{
     MEMBER M;
     char m1code[10],mname[26],mphone[10],maddress[33],ch;
     int m2code=0,mcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,7);cout<<"Enter Code no. of the Member to be modified";
               gotoxy(5,8);cout<<"             or                  ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.........: ";
               gets(m1code);
               m2code=atoi(m1code);
               mcode=m2code;
               if(m1code[0]=='0')return;
               if(strlen(m1code)==0)
               M.list();
               else
               break;
          }
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);cout<<"\7Enter Correctly";
               getch();
          }
          if(valid && !memfound(mcode))
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
               gotoxy(5,14);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     MEMBER::display(mcode);
     do
     {
          gotoxy(5,10);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;
gotoxy(5,13);cout<<"Name  :";
gotoxy(5,14);cout<<"Phone  :";
gotoxy(5,15);cout<<"Address:";
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the name of the member or <ENTER> for no change";
     gotoxy(19,13);clreol();
     gets(mname);
     strupr(mname);
     if(mname[0]=='0')return;
     if(strlen(mname) > 25)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly(Range: 1..25)";
          getch();
     }
}while(!valid);
if(strlen(mname)==0)
strcpy(mname,memname(mcode));
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the Phone no. of Member or <ENTER> for no change";
     gotoxy(19,14);clreol();
     gets(mphone);
     if(mphone[0]=='0')return;
     if((strlen(mphone) < 7 && strlen(mphone) > 0 || (strlen(mphone) > 9)))
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7Enter Correctly";
          getch();
     }
}while(!valid);
if(strlen(mphone)==0)
strcpy(mphone,memphone(mcode));
do
{
     valid=1;
     gotoxy(5,25);clreol();
     cout<<"Enter the address of the member or <ENTER> for no change";
     gotoxy(19,15);clreol();
     gets(maddress);
     strupr(maddress);
     if(maddress[0]=='0')return;
     if(strlen(maddress) > 32)
     {
          valid=0;
          gotoxy(5,25);clreol();
          cout<<"\7 Enter correctly(Range: 1..32)";
```

```
               getch();
          }
     }while(!valid);
     if(strlen(maddress)==0)
     strcpy(maddress,memaddress(mcode));
     gotoxy(5,25);clreol();
     do
     {
          gotoxy(5,18);clreol();
          cout<<"Do you want to save changes(y/n)";
          ch=getche();
          ch=toupper(ch);
          if(ch=='0')return;
     }while(ch!='Y' && ch!='N');
     if(ch=='N')
     return;
     MEMBER::modify(mcode,mname,mphone,maddress);
     gotoxy(5,23);
     cout<<"\7Record Modified";
     getch();
}
//gives book code to delete book record
void WORKING::deletebook(void)
{
     BOOK B;
     char t1code[5],tname[33],tauthor[26],ch;
     int t2code=0,tcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,5);cout<<"Enter Code or Name of the Book to be deleted";
               gotoxy(5,6);cout<<"             or                  ";
               gotoxy(5,7);cout<<"Press <ENTER> for help.........: ";
               gets(t1code);
               if(t1code[0]=='0')return;
               if(strlen(t1code)==0)
               B.list();
               else
               break;
          }
          t2code=atoi(t1code);
          tcode=t2code;
          if((tcode==0 && !booknamefound(t1code)) || (tcode!=0 && !
bookfound(tcode)))
          {
               valid=0;
               gotoxy(5,10);cout<<"\7Record not found";
               gotoxy(5,11);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
```

```
     if(tcode==0)
     tcode=bookcodeof(t1code);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     BOOK::display(tcode);
     do
     {
          gotoxy(5,13);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;
     int tavail,tcopies;
     tavail=available(tcode);
     tcopies=noofcopies(tcode);
     if(tavail!=tcopies)
     {
          gotoxy(5,15);cout<<"\7Record cannot be deleted.This book is issued.";
          gotoxy(5,16);cout<<"\7Type special code to delete";
          if(getch()!='7')
          {        gotoxy(5,19);cout<<"\7*";
                getch();
                return;
          }
          else
                gotoxy(5,19);cout<<"\7*";
     }
     BOOK::deleterec(tcode);
     gotoxy(5,23);cout<<"\7Record Deleted";
     getch();
}
//gives mem. code to delete member record
void WORKING::deletemem(void)
{
     MEMBER M;
     char m1code[10],mname[26],mphone[10],maddress[33],ch;
     int m2code=0,mcode=0;
     int valid;
     do
     {
          valid=1;
          while(1)
          {
               clrscr();
               gotoxy(72,1);cout<<"<0>=Exit";
               gotoxy(5,7);cout<<"Enter Code no. of the Member to be deleted";
               gotoxy(5,8);cout<<"             or                 ";
               gotoxy(5,9);cout<<"Press <ENTER> for help.....: ";
               gets(m1code);
               m2code=atoi(m1code);
               mcode=m2code;
               if(m1code[0]=='0')return;
               if(strlen(m1code)==0)
               M.list();
               else
               break;
```

```
          }
          if(mcode==0)
          {
               valid=0;
               gotoxy(5,25);cout<<"\7Enter Correctly";
               getch();
          }
          if(valid && !memfound(mcode))
          {
               valid=0;
               gotoxy(5,13);cout<<"\7Record not found";
               gotoxy(5,14);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch=getch();
               if(ch==27)return;
          }
     }while(!valid);
     clrscr();
     gotoxy(72,1);cout<<"<0>=Exit";
     MEMBER::display(mcode);
     do
     {
          gotoxy(5,10);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;
     if(issued(mcode))
     {
          gotoxy(5,15);
          cout<<"\7Record cannot be deleted.Member has a book.";
          getch();
          return;
     }
     MEMBER::deleterec(mcode);
     gotoxy(5,23);cout<<"\7Record Modified";
     getch();
}
//Main func. calling introduction and main menu
void main()
{
     MENU menu;
     menu.introduc();
     menu.mainmenu();
}
```