Project Report on

NUMBER EATERS (A Mental Game for Kids)

Developed & Submitted By

Vivek

(Project Level Computer Programming Couse)

Under the Guidance of

Mr. Ashok Bhardwaj

(Director, Bharti Computer Education)

For

Bharti Computer Education
1203/1, Near Shri Krishna Mandir, Najafgarh, New Delhi- 110043

BHARTI COMPUTER EDUCATION

1203/1, Near Shri Krishna Mandir, Najafgarh, New Delhi- 110043
Ph. 9212187612, 8744007612

CERTIFICATE

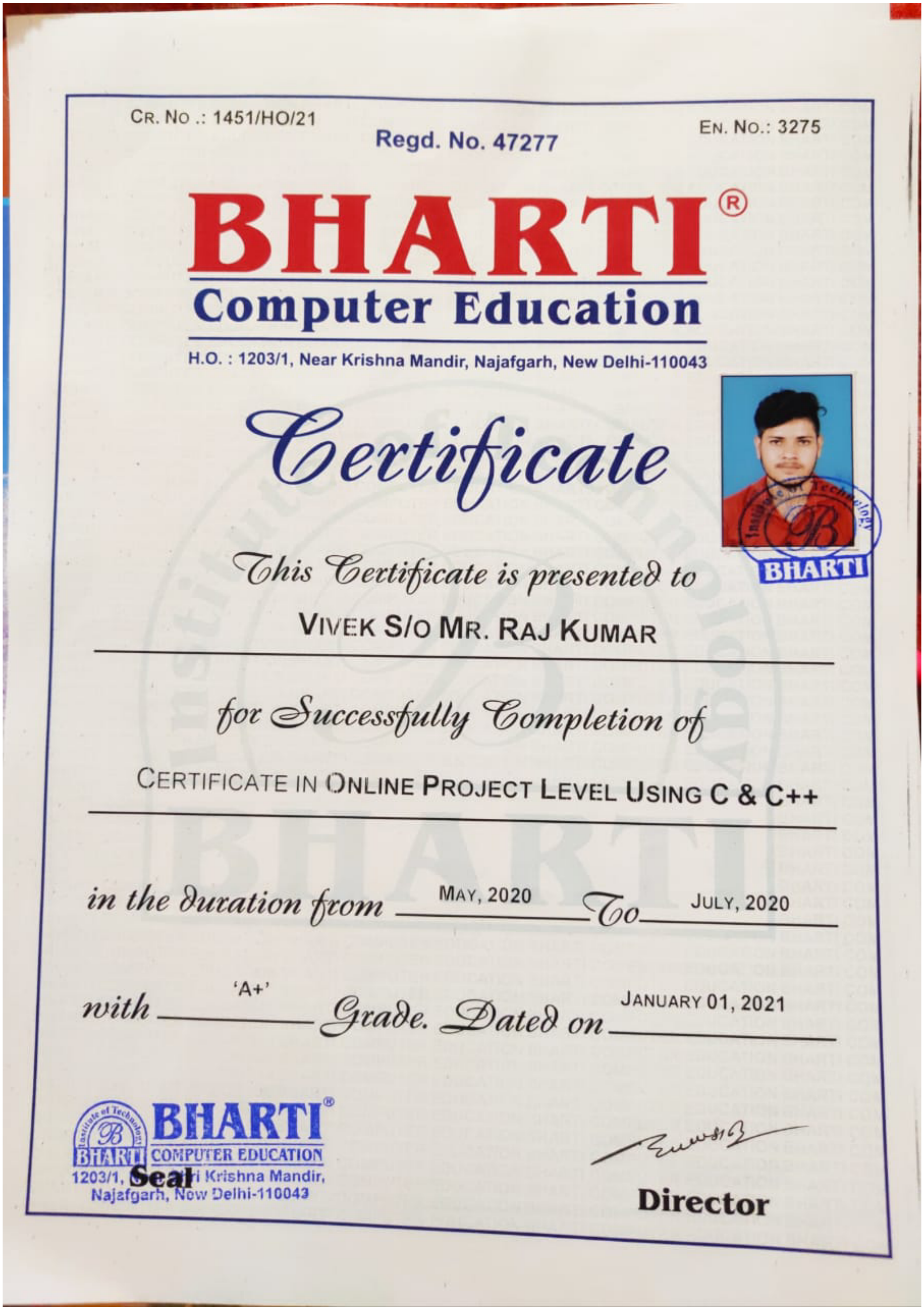
This is to certify that VIVEK s/o Mr. Raj Kumar, student of Computer Programming Course of Project Level has prepared the report on the Project entitled *“STUDENT DATABASE PROJECT”.*

The report is the result of his efforts & endeavors. The report is found worthy of acceptance as final project report for the Computer Programming Course for Project Level.

He has prepared the report under my guidance and I wish him all the best for his future life.

Seal ASHOK
BHARDWAJ

(B.Com, MCA, M.Sc.(IT & CA))
Director



CR. NO.: 1451/HO/21

Regd. No. 47277

EN. NO.: 3275

BHARTI®

Computer Education

H.O. : 1203/1, Near Krishna Mandir, Najafgarh, New Delhi-110043

Certificate

Institute of Technology
BHARTI

This Certificate is presented to

VIVEK S/O MR. RAJ KUMAR

for Successfully Completion of

CERTIFICATE IN ONLINE PROJECT LEVEL USING C & C++

in the duration from MAY, 2020 *To* JULY, 2020

with 'A+' *Grade. Dated on* JANUARY 01, 2021

Institute of Technology
BHARTI®
COMPUTER EDUCATION
1203/1, Shri Krishna Mandir,
Najafgarh, New Delhi-110043

Seal

Director

PROJECT REPORT

CONTENTS

INTRODUCTION

This project (*STUDENT DATABASE PROJECT*) is written in C language. This project allows you input any course along with its details, student details and fee details.

Whenever any student wants to enroll in any course then he/she get complete information about the course by just entering the course ID.

Following project allows you to access

1. all the database of course .
2. all the database of student .
3. all the database of fee .

You are also able to get details of those students whom LEFT the course, COMPLETE the course and those whom are in the ACTIVE course.

OBJECTIVE

- This project in C is a simple console application with no computer graphics.
- The Project is specially designed to handle database and it is a type of mental exercise.
- There comes a window in the project in which you have to enter details in course, student and fee database.
- Developing this project helps to reinforce many of the C and programming concepts we have met already.
- This project provide a valuable experience of design and implementation of a large program.
- It also provide a framework for a more challenging and thus rewarding, laboratory exercise.

REQUIREMENTS

1. Software Requirements:
   - Operating system – Windows & MS-DOS
   - Application software – Turbo C++
   - Language – C language

2. Hardware Requirements:
   - RAM- 2GB
   - Hard disk- 512MB
   - Processor –Any Intel processor

TECHNOLOGY USED

1. Turbo C

   Turbo C version 2.0 is used in this game. Turbo C is a discontinued Integrated Development Environment and compiler for the C programming language from Borland. First introduced in 1987, it was noted for its integrated development environment, small size, fast compile speed, comprehensive manuals and low price .In May 1990, Borland replaced Turbo C with Turbo C++.

2. C Language

   C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. By design, C provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as various application software for computers ranging from supercomputers to embedded systems.

   C was originally developed by Dennis Ritchie between 1969 and 1973 at Bell Labs, and used to re-implement the Unix operating system. It has since become one of the most widely used programming languages of all time.

   C is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming

SIGNIFICANCE OF PROJECT IN LIFE

- Practical skills
- Strong memory
- Power to decide
- Idea how to behave in different environment
- Powerful tool to help children to develop certain life skiils
- Gain self confidence
- Improves hand and eye coordination
- For motivational challenges.

FUTURE SCOPE OF PROJECT

- Use Project for skills based learning
- Important as learning point of view for children
- Multilayer feature can be added.
- Virtual reality

REFERENCES

- Guided by Mr. Ashok Bhardwaj (Director, Bharti Computer Education)
- Books

  Let's play with C by Mr. Ashok Bhardwaj

CONCLUSION:

We had successfully made this PROJECT in C language which store data of various courses, students and fee and displayed required details on screen.

SOURCE CODE:

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<stdio.h>
#include<string.h>
#include<dos.h>
#include<ctype.h>
#include<graphics.h>
// Declaration of course class
class course
{
    private:
                char cn[20],dur[20];
                float fee;
                int cid;
    public:
                void getCourseDetail(int);
                void showCourseDetail(int);
                void showDetail();
                int returnCourseFee();
                char* returnName();
                int returnCid();
};
// Declaration of student class
class student
{
    private:
                int cid1,roll;
                char n[20],fn[20],dob[20],add[20],status[20];
    public:
                int getStudentDetail(int);
                void showStudentDetail(int);
                int showName();
                void setStatus(char []);
                char* returnName1();
                char* returnStatus();
                int returnCid1();
                int returnRoll();
};
// Declaration of fee class
class fee
{
   private:
                int feeRoll,recieptNo;
                float amount;
                char date[20];
```

```
  public:
          int returnRecieptNo()
          {
               return recieptNo;
          }
          int returnAmount()
          {
               return amount;
          }
          int returnFeeRoll()
          {
               return feeRoll;
          }
          void getFeeDetail(int a,int b,char c[],float d)
          {
               feeRoll=a;
               recieptNo=b;
               strcpy(date,c);
               amount=d;
          }
};
// Introduction slide function
void starting()
{
  int gd=0,gm;
  initgraph(&gd,&gm," ");
  settextstyle(10,0,4);
  setcolor(4);
  outtextxy(100,50,"* DATABASE *");
  settextstyle(6,0,4);
  setcolor(2);
  delay(200);
  outtextxy(50,150,"* A project to store database of  ");
  outtextxy(70,200,"students.");
  setcolor(14);
  delay(200);
  outtextxy(50,250,"* Special thanks to my ideal - ");
  setcolor(4);
  settextstyle(10,0,3);
  delay(200);
  outtextxy(200,300,"MR. ASHOK BHARDWAJ");
  setcolor(15);
  settextstyle(6,0,4);
  delay(200);
  outtextxy(50,360,"* A project by - Vivek (IT)");
  setcolor(3);
  delay(200);
  outtextxy(50,410,"* Press any key to START the Project");
  delay(1500);
  getch();
  cleardevice();
  settextstyle(10,0,4);
  setcolor(4);
  outtextxy(70,50,"* Basic Controls- *");
```

```
    settextstyle(6,0,4);
    setcolor(2);
    delay(200);
    outtextxy(50,150,"* Please wait getting you to the MAIN MENU. ");
    setcolor(9);
    delay(200);
    outtextxy(50,250,"* Press 4 to exit the database. ");
    setcolor(14);
    outtextxy(50,350,"* Press any other key get in the database.");
    delay(2500);
    cleardevice();
    closegraph();
}
// Ending slide function
void end()
{
    int gd=0,gm;
    initgraph(&gd,&gm," ");
    settextstyle(10,0,4);
    setcolor(4);
    outtextxy(100,50,"* DATABASE *");
    settextstyle(6,0,4);
    setcolor(2);
    outtextxy(50,150,"* Thanks for using this DATABASE Project. ");
    setcolor(3);
    outtextxy(50,200,"* I wish it helps you.");
    setcolor(14);
    outtextxy(50,250,"* Special thanks to my ideal - ");
    setcolor(4);
    settextstyle(10,0,3);
    outtextxy(200,300,"MR. ASHOK BHARDWAJ");
    setcolor(15);
    settextstyle(6,0,4);
    outtextxy(50,360,"* A project by - Vivek (IT)");
    delay(2500);
}
// Box function
void box()
{
   int a=201,b=187,c=188,d=200,e=205,f=186,g=204,h=185,i=203,j=202,row=1,col=1;
   char ch;
   for(col=1;col<80;col++)
   {
       gotoxy(col,1);
       ch=e;
       cout<<ch;
   }
   for(;row<6;row++)
   {
       gotoxy(80,row);
       ch=f;
       cout<<ch;
   }
   for(;row<24;row++)
```

```
{
    gotoxy(80,row);
    ch=f;
    cout<<ch;
}
for(;col>1;col--)
{
    gotoxy(col,24);
    ch=e;
    cout<<ch;
}
for(;row>6;row--)
{
    gotoxy(1,row);
    ch=f;
    cout<<ch;
}
for(;row>1;row--)
{
    gotoxy(1,row);
    ch=f;
    cout<<ch;
}
for(;col<80;col++)
{
    gotoxy(col,6);
    ch=e;
    cout<<ch;
}
gotoxy(1,1);
ch=a;
cout<<ch;
gotoxy(80,1);
ch=b;
cout<<ch;
gotoxy(80,24);
ch=c;
cout<<ch;
gotoxy(1,24);
ch=d;
cout<<ch;
gotoxy(80,6);
ch=h;
cout<<ch;
gotoxy(1,6);
ch=g;
cout<<ch;
gotoxy(30,2);
cout<<"BHARTI COMPUTER EDUCATION";
gotoxy(17,3);
cout<<"1203/1,Near Shri Krishna Mandir,Najafgarh,N.D.-43";
for(row=6;row<24;row++)
{
   gotoxy(22,row);
```

```
    ch=f;
    cout<<ch;
  }
  for(row=6;row<24;row++)
  {
    gotoxy(60,row);
    ch=f;
    cout<<ch;
  }
  gotoxy(60,6);
  ch=i;
  cout<<ch;
  gotoxy(60,24);
  ch=j;
  cout<<ch;
  gotoxy(22,6);
  ch=i;
  cout<<ch;
  gotoxy(22,24);
  ch=j;
  cout<<ch;

}
// editBox function to edit box parameters
void edit()
{
  int e=205,row=1;
  char ch;
  for(row=6;row<=24;row++)
  {
     gotoxy(22,row);
     cout<<" ";
  }
  for(row=6;row<=24;row++)
  {
     gotoxy(60,row);
     cout<<" ";
  }
  for(int col=1;col<80;col++)
  {
    gotoxy(col,8);
    ch=e;
    cout<<ch;
  }
  gotoxy(1,8);
  ch=204;
  cout<<ch;
  gotoxy(80,8);
  ch=185;
  cout<<ch;
  gotoxy(22,6);
  ch=e;
  cout<<ch;
  gotoxy(60,6);
  ch=e;
  cout<<ch;
  gotoxy(22,24);
  ch=e;
  cout<<ch;
  gotoxy(60,24);
  ch=e;
  cout<<ch;
}
// smallbox function
void smallbox()
{
  int a=201,e=205,f=186,g=185,j=202,col=1;
  char ch;
  for(col=45;col<80;col++)
  {
    gotoxy(col,22);
    ch=e;
    cout<<ch;
  }
  gotoxy(80,22);
  ch=g;
  cout<<ch;
  gotoxy(45,23);
  ch=f;
  cout<<ch;
  gotoxy(45,22);
  ch=a;
  cout<<ch;
  gotoxy(45,24);
  ch=j;
  cout<<ch;
}
// getCourseDetail function for inputing course details
void course::getCourseDetail(int id)
{
   cid=id;
   gotoxy(12,11);
   cout<<"Course ID:- ";
   gotoxy(12,13);
   cout<<"Course Name:- ";
   gotoxy(12,15);
   cout<<"Duration:- ";
   gotoxy(12,17);
   cout<<"Fee:-  ";
   gotoxy(32,11);
   cout<<cid;
   gotoxy(32,13);
   gets(cn);
   gotoxy(32,15);
   gets(dur);
   gotoxy(32,17);
   cin>>fee;
}
```

```
// showCourseDetail function for showing complete course details
void course::showCourseDetail(int row)
{

    gotoxy(7,7);
    cout<<"C-ID";
    gotoxy(18,7);
    cout<<"Course Name";
    gotoxy(39,7);
    cout<<"Duration";
    gotoxy(66,7);
    cout<<"Fee";
    gotoxy(8,row);
    cout<<cid;
    gotoxy(20,row);
    cout<<strupr(cn);
    gotoxy(42,row);
    cout<<strupr(dur);
    gotoxy(65,row);
    cout<<fee;
}
// showDetail function for showing some details
void course::showDetail()
{
    gotoxy(50,10);
    cout<<"Course Name";
    gotoxy(50,12);
    cout<<"Duration";
    gotoxy(50,14);
    cout<<"Fee";
    gotoxy(70,10);
    cout<<strupr(cn);
    gotoxy(70,12);
    cout<<strupr(dur);
    gotoxy(70,14);
    cout<<fee;

}
// returnName function for returning course name
char* course::returnName()
{
    return cn;
}
// returnCid function for returning courseId
int course::returnCid()
{
   return cid;

}
// returnCourseFee for returning courseFee
int course::returnCourseFee()
{
   return fee;
}
```

```
// getStudentDetail function for inputing student details
int student::getStudentDetail(int rol)
{
  roll=rol;
  gotoxy(32,7);
  cout<<"STUDENT DETAIL";
  gotoxy(5,10);
  cout<<"Roll No.:-";
  gotoxy(5,12);
  cout<<"Student Name:-";
  gotoxy(5,14);
  cout<<"Father's Name:-";
  gotoxy(5,16);
  cout<<"Date of Birth:-";
  gotoxy(5,18);
  cout<<"Address:-";
  gotoxy(5,20);
  cout<<"Course ID:-";
  gotoxy(25,10);
  cout<<roll;
  gotoxy(25,12);
  gets(n);
  gotoxy(25,14);
  gets(fn);
  gotoxy(25,16);
  gets(dob);
  gotoxy(25,18);
  gets(add);
  gotoxy(25,20);
  cin>>cid1;
  return cid1;

}
// showStudentDetail function for showing complete student details
void student::showStudentDetail(int row)
{
   gotoxy(3,7);
   cout<<"R NO.";
   gotoxy(12,7);
   cout<<"C-ID";
   gotoxy(21,7);
   cout<<"Student Name";
   gotoxy(37,7);
   cout<<"Father's Name";
   gotoxy(55,7);
   cout<<"Status";
   gotoxy(65,7);
   cout<<"Course Name";
   gotoxy(5,row);
   cout<<roll;
   gotoxy(13,row);
   cout<<cid1;
   gotoxy(23,row);
```

```
    strupr(n);
    cout<<n;
    gotoxy(39,row);
    strupr(fn);
    cout<<fn;
    gotoxy(54,row);
    strupr(status);
    cout<<status;
    gotoxy(67,row);

}
// returnStatus function for returning status of student profile
char* student::returnStatus()
{
    return status;
}
// returnName1 function for returning name of student
char* student::returnName1()
{
    return n;
}
// returnCid1 function for returning courseId of student
int student::returnCid1()
{
    return cid1;
}
// returnRoll function for returning roll number of student
int student::returnRoll()
{
    return roll;
}
// setStatus function for setting the status
void student::setStatus(char b[])
{
    strcpy(status,b);
}
// currentRecieptNo function for calculating Reciept Number
int currentRecieptNo()
{
    fee objFee;
    int rpt=0,rec=0;
    fstream fin;
    fin.open("fee1.dat",ios::in);
    fin.seekg(0,ios::end);
    rec=fin.tellg()/sizeof(objFee);
    fin.seekg(0,ios::beg);
    for(int i=1;i<=rec;i++)
    {
            fin.read((char*)&objFee,sizeof(objFee));
            if(objFee.returnRecieptNo() > rpt)
            {
                rpt=objFee.returnRecieptNo();
            }
    }
```

```
   fin.close();
   rpt++;
   return rpt;
}
// totalFee function for returning total fee for a particular Roll Number
float totalFee(int roll)
{
   fee objFee;
   int rec=0,i;
   float total=0;
   fstream fin;
   fin.open("fee1.dat",ios::in);
   fin.seekg(0,ios::end);
   rec=fin.tellg()/sizeof(objFee);
   fin.seekg(0,ios::beg);
   for(i=1;i<=rec;i++)
   {
     fin.read((char*)&objFee,sizeof(objFee));
     if(objFee.returnFeeRoll()==roll)
     {
             total+=objFee.returnAmount();
     }
   }
   fin.close();
   return total;
}
// courseFee function for returning Course Fee
int courseFee(int roll)
{
   course objCourse;
   student objStudent;
   fstream fin;
   int i,rec=0,courseFee=0,courseID=0;
   fin.open("stu1.dat",ios::in);
   fin.seekg(0,ios::end);
   rec=fin.tellg()/sizeof(objStudent);
   fin.seekg(0,ios::beg);
   for(i=1;i<=rec;i++)
   {
           fin.read((char*)&objStudent,sizeof(objStudent));
           if(objStudent.returnRoll()==roll)
           {
              courseID=objStudent.returnCid1();
              gotoxy(32,16);
              cout<<objStudent.returnName1();
           }
   }
   fin.close();
   fin.open("course1.dat",ios::in);
   fin.seekg(0,ios::end);
   rec=fin.tellg()/sizeof(objCourse);
   fin.seekg(0,ios::beg);
   for(i=1;i<=rec;i++)
   {
           fin.read((char*)&objCourse,sizeof(objCourse));
           if(objCourse.returnCid()==courseID)
           {
              courseFee=objCourse.returnCourseFee();
              gotoxy(32,18);
              cout<<objCourse.returnName();
           }
   }
   fin.close();
   return courseFee;
}
// MAIN FUNCTION
void main()
{
   // Calling Starting function
   starting();
   clrscr();
   // Declaring variables of character type
   char ch1,ch2,ch3,ch4,name[20],c,x[20];
   //  Declaring variables of integer type
   int i,j,rec,rec1,rec2,count,row,id=0,rollno=0;
   // Creating objects of fstream
   fstream fin,fin1,fin2;
   // Creating object of Course class
   course objCourse;
   // Creating object of Student class
   student objStudent;
   // Creating object of Fee class
   fee objFee;
   do
   {
            clrscr();
            box();
            gotoxy(35,5);
            cout<<"MAIN MENU";
            gotoxy(32,9);
            cout<<"1. Course Detail";
            gotoxy(32,11);
            cout<<"2. Student Detail";
            gotoxy(32,13);
            cout<<"3. Fee Detail";
            gotoxy(32,15);
            cout<<"4. Exit";
            gotoxy(32,17);
            cout<<"Enter your choice:- ";
            ch1=getche();
            switch(ch1)
            {
                      // **INSIDE COURSE DATABASE**
                case '1':
                            do
                            {
                                  clrscr();
                                  box();
```

```
gotoxy(32,5);
cout<<"COURSE DETAIL";
gotoxy(29,9);
cout<<"1. Add Course";
gotoxy(29,11);
cout<<"2. Modify Course Detail";
gotoxy(29,13);
cout<<"3. Delete Course";
gotoxy(29,15);
cout<<"4. Display Course List";
gotoxy(29,17);
cout<<"5. Return To Main Menu";
gotoxy(29,19);
cout<<"Enter your choice:- ";
ch2=getche();
switch(ch2)
{
  case '1':
            do
            {
              //**OPENING FILE course1 IN READING MODE TO CALCULATE COURSE ID**
              fin.open("course1.dat",ios::in);
              fin.seekg(0,ios::end);
              rec=fin.tellg()/sizeof(objCourse);
              fin.seekg(0,ios::beg);
              for(i=1;i<=rec;i++)
              {
                    fin.read((char*)&objCourse,sizeof(objCourse));
                    if(objCourse.returnCid() >id)
                    {
                       id=objCourse.returnCid();
                    }
              }
              fin.close();
              clrscr();
              box();
              edit();
              gotoxy(32,7);
              cout<<"COURSE DETAIL";
              //**OPENING FILE course1 IN APPEND MODE TO ADD COURSE**
              fin.open("course1.dat",ios::app);
              {
                     objCourse.getCourseDetail(++id);
                     fin.write((char*)&objCourse,sizeof(objCourse));
                     gotoxy(29,19);
                     cout<<"Do You Want To Continue:- ";
                     ch3=getche();
              }
              fin.close();
             }while(ch3!='n');
             break;
  case '2':
             clrscr();
             box();
```

```
edit();
        //**PRINTING ALL COURSES ON SCREEN**
fin.open("course1.dat",ios::in);
fin.seekg(0,ios::end);
rec=fin.tellg()/sizeof(objCourse);
fin.seekg(0,ios::beg);
count=1;
for(i=1,row=10;i<=rec;i++)
{
         fin.read((char*)&objCourse,sizeof(objCourse));
         objCourse.showCourseDetail(row++);
         count++;
         if(count==14)
         {
           getch();
           clrscr();
           box();
           edit();
           row=10;
           count=1;
         }
}
fin.close();
getch();
        //**OPENING course1 FILE IN READ MODE AND temp FILE IN WRITE MODE**
fin.open("course1.dat",ios::in);
fin1.open("temp.dat",ios::out);
fin.seekg(0,ios::end);
rec=fin.tellg()/sizeof(objCourse);
fin.seekg(0,ios::beg);
gotoxy(25,21);
        //**TAKING COURSE ID FOR MODIFYING COURSE DETAIL**
cout<<"Enter the course id to modify:- ";
cin>>id;
clrscr();
box();
edit();
        //**UPDATING COURSE DETAILS**
for(i=1;i<=rec;i++)
{
        fin.read((char*)&objCourse,sizeof(objCourse));
        if(objCourse.returnCid()==id)
        {

           objCourse.getCourseDetail(id);
           fin1.write((char*)&objCourse,sizeof(objCourse));
        }
        else
           fin1.write((char*)&objCourse,sizeof(objCourse));
}
fin.close();
fin1.close();
remove("course1.dat");
```

```
                        rename("temp.dat","course1.dat");
                        break;
                case '3':
                        clrscr();
                        box();
                        edit();
                            //**PRINTING ALL COURSES ON SCREEN**
                        fin.open("course1.dat",ios::in);
                        fin.seekg(0,ios::end);
                        rec=fin.tellg()/sizeof(objCourse);
                        fin.seekg(0,ios::beg);
                        count=1;
                        for(i=1,row=10;i<=rec;i++)
                        {
                            fin.read((char*)&objCourse,sizeof(objCourse));
                            objCourse.showCourseDetail(row++);
                            count++;
                            if(count==14)
                            {
                              getch();
                              clrscr();
                              box();
                              edit();
                              row=10;
                              count=1;
                            }
                        }
                        fin.close();
                        getch();
                            //**OPENING course1 FILE IN READ MODE AND temp FILE IN WRITE
MODE**
                        fin.open("course1.dat",ios::in);
                        fin1.open("temp.dat",ios::out);
                        gotoxy(27,21);
                            //**TAKING COURSE ID TO DELETE COURSE**
                        cout<<"Enter course id to delete:-";
                        cin>>id;
                        fin.seekg(0,ios::end);
                        rec=fin.tellg()/sizeof(objCourse);
                        fin.seekg(0,ios::beg);
                            //**DELETING COURSE**
                        for(i=1;i<=rec;i++)
                        {
                            fin.read((char*)&objCourse,sizeof(objCourse));
                            if(objCourse.returnCid()!=id)
                            {
                              fin1.write((char*)&objCourse,sizeof(objCourse));
                            }
                        }
                        fin.close();
                        fin1.close();
                        remove("course1.dat");
                        rename("temp.dat","course1.dat");
                        break;
```

```
                    case '4':
                            clrscr();
                            box();
                            edit();
                                //**DISPLAYING ALL COURSES**
                            fin.open("course1.dat",ios::in);
                            fin.seekg(0,ios::end);
                            rec=fin.tellg()/sizeof(objCourse);
                            fin.seekg(0,ios::beg);
                            row=9;
                            for(i=1;i<=rec;i++)
                            {
                                fin.read((char*)&objCourse,sizeof(objCourse));
                                objCourse.showCourseDetail(row++);

                            }
                            fin.close();
                            getch();
                            break;
                    case '5':
                            break;
                    }
                }while(ch2!='5');
                break;
            // **INSIDE STUDENT DATABASE**
        case '2':
                do
                {
                    clrscr();
                    box();
                    gotoxy(32,5);
                    cout<<"Student Detail";
                    gotoxy(29,9);
                    cout<<"1. Name Entry";
                    gotoxy(29,11);
                    cout<<"2. Modify Detail";
                    gotoxy(29,13);
                    cout<<"3. Course Complete";
                    gotoxy(29,15);
                    cout<<"4. Course Left";
                    gotoxy(29,17);
                    cout<<"5. Show All Addmission";
                    gotoxy(29,19);
                    cout<<"6. Return To Main Menu";
                    gotoxy(29,21);
                    cout<<"Enter your choice:-";
                    ch2=getche();
                    switch(ch2)
                    {
                     case '1': do
                            {
                                      //**OPENING FILE stu1 IN READING MODE TO CALCULATE
ROLL NO**
                                  fin.open("stu1.dat",ios::in);
fin.seekg(0,ios::end);
rec=fin.tellg()/sizeof(objStudent);
fin.seekg(0,ios::beg);
for(i=1;i<=rec;i++)
{
    fin.read((char*)&objStudent,sizeof(objStudent));
    if(objStudent.returnRoll() > rollno)
    {
              rollno=objStudent.returnRoll();
    }
}
fin.close();
clrscr();
box();
edit();
gotoxy(32,7);
cout<<"STUDENT DETAIL";
            //**OPENING FILE stu1 IN APPEND MODE TO ADD STUDENT
DETAIL**

fin.open("stu1.dat",ios::app);
{
    id=objStudent.getStudentDetail(++rollno);
    objStudent.setStatus("ACTIVE");
    if(isdigit(id)==0 && id<=47)
    {
            fin1.open("course1.dat",ios::in);
            fin1.seekg(0,ios::end);
            rec1=fin1.tellg()/sizeof(objCourse);
            fin1.seekg(0,ios::beg);
            for(i=1;i<=rec1;i++)
            {
              fin1.read((char*)&objCourse,sizeof(objCourse));
              if(objCourse.returnCid()==id)
              {
                 objCourse.showDetail();
                 break;
              }
              else if(objCourse.returnCid()!=id && i==rec1)
              {
                 gotoxy(50,13);
                 cout<<" Course do not exist";
              }
            }
    }
    else
    {
              gotoxy(50,13);
              cout<<" Course do not exist";
    }
    smallbox();
    gotoxy(46,23);
    cout<<"Do you want to add course (y/n):- ";
    c=getche();
    if(c=='y' || c=='Y')
```

```
                                            {
                                                    fin.write((char*)&objStudent,sizeof(objStudent));
                                            }
                                            else
                                            {
                                                    rollno--;
                                            }
                                            gotoxy(10,23);
                                            cout<<"Do You Want To Continue:- ";
                                            ch3=getche();
                                        }
                                        fin.close();
                                        fin1.close();
                                }while(ch3!='n');
                                break;
                        case '2':
                                clrscr();
                                box();
                                edit();
                                fin.open("stu1.dat",ios::in);
                                fin.seekg(0,ios::end);
                                rec=fin.tellg()/sizeof(objStudent);
                                fin.seekg(0,ios::beg);
                                count=1;
                                                    //**PRINTING ALL STU_DETAILS WITH COURSE NAME ON
SCREEN**
                                for(i=1,row=10;i<=rec;i++)
                                {
                                        fin.read((char*)&objStudent,sizeof(objStudent));
                                        objStudent.showStudentDetail(row++);
                                        fin1.open("course1.dat",ios::in);
                                        fin1.seekg(0,ios::end);
                                        rec1=fin1.tellg()/sizeof(objCourse);
                                        fin1.seekg(0,ios::beg);
                                        for(j=1;j<=rec1;j++)
                                        {
                                          fin1.read((char*)&objCourse,sizeof(objCourse));
                                          if(objStudent.returnCid1()==objCourse.returnCid())
                                          {
                                                  strcpy(x,objCourse.returnName());
                                                  cout<<strupr(x);
                                          }
                                        }
                                        fin1.close();
                                        count++;
                                        if(count==14)
                                        {
                                          getch();
                                          clrscr();
                                          box();
                                          edit();
                                          row=10;
                                          count=1;
                                        }
                                }
                                fin.close();
                                getch();
                                            //**OPENING stu1 FILE IN READING AND temp FILE IN WRITE
MODE**
                                fin.open("stu1.dat",ios::in);
                                fin1.open("temp.dat",ios::out);
                                fin.seekg(0,ios::end);
                                rec=fin.tellg()/sizeof(objStudent);
                                fin.seekg(0,ios::beg);
                                gotoxy(25,21);
                                            //**TAKING ROLL NUMBER TO MODIFY STUDENT DETAILS**
                                cout<<"Enter the roll no. to modify:-";
                                cin>>rollno;
                                clrscr();
                                box();
                                edit();
                                            //**MODIFYING STUDENT DETAILS**
                                for(i=1;i<=rec;i++)
                                {
                                    fin.read((char*)&objStudent,sizeof(objStudent));
                                    if(objStudent.returnRoll()==rollno &&
objStudent.returnStatus()=="ACTIVE")
                                    {

                                      objStudent.getStudentDetail(rollno);
                                      fin1.write((char*)&objStudent,sizeof(objStudent));
                                    }
                                    else
                                      fin1.write((char*)&objStudent,sizeof(objStudent));
                                }
                                fin.close();
                                fin1.close();
                                remove("stu1.dat");
                                rename("temp.dat","stu1.dat");
                                break;
                        case '3':
                                 do
                                 {
                                      clrscr();
                                      box();
                                      gotoxy(32,5);
                                      cout<<"Student Detail";
                                      gotoxy(29,9);
                                      cout<<"1. Course Complete ";
                                      gotoxy(29,11);
                                      cout<<"2. Display CC List ";
                                      gotoxy(29,13);
                                      cout<<"3. Exit";
                                      gotoxy(29,18);
                                      cout<<" Enter your choice:-";
                                      ch4=getche();
                                      switch(ch4)
                                      {
```

```
                              case '1':
                                    clrscr();
                                    box();
                                    edit();
                                    gotoxy(32,5);
                                        //**PRINTING STUDENTS DETAILS ALONG
WITH COURSE NAME**
                                    cout<<"Students Detail";
                                    fin.open("stu1.dat",ios::in);
                                    fin.seekg(0,ios::end);
                                    rec=fin.tellg()/sizeof(objStudent);
                                    fin.seekg(0,ios::beg);
                                    row=10;
                                    for(i=1;i<=rec;i++)
                                    {
                                     fin.read((char*)&objStudent,sizeof(objStudent));
                                     objStudent.showStudentDetail(row++);
                                     fin1.open("course1.dat",ios::in);
                                     fin1.seekg(0,ios::end);
                                     rec1=fin1.tellg()/sizeof(objCourse);
                                     fin1.seekg(0,ios::beg);
                                     for(j=1;j<=rec1;j++)
                                     {
                                      fin1.read((char*)&objCourse,sizeof(objCourse));
                                      if(objStudent.returnCid1()==objCourse.returnCid())
                                      {
                                        strcpy(x,objCourse.returnName());
                                        cout<<strupr(x);
                                      }
                                     }
                                     fin1.close();
                                    }
                                    fin.close();
                                    gotoxy(10,21);
                                        //**TAKING ROLL NUMBER TO MARK HIS
COURSE COMPLETED**
                                    cout<<"Enter the rollno of student whose course is

completed:- ";

                                    cin>>rollno;
                                    fin.open("stu.dat",ios::in);
                                    fin1.open("cc.dat",ios::out);
                                    fin.seekg(0,ios::end);
                                    rec=fin.tellg()/sizeof(objStudent);
                                    fin.seekg(0,ios::beg);
                                        //**MARKING GIVEN ROLL NUMBER AS
COURSE COMPLETED**
                                    for(i=1;i<=rec;i++)
                                    {
                                     fin.read((char*)&objStudent,sizeof(objStudent));
                                     if(objStudent.returnRoll()==rollno)
                                     {
                                       objStudent.setStatus("C_Comp");
                                       fin1.write((char*)&objStudent,sizeof(objStudent));
```

```
                      }
                      else
                      {
                          fin1.write((char*)&objStudent,sizeof(objStudent));
                      }
                    }
                    fin.close();
                    fin1.close();
                    remove("stu1.dat");
                    rename("cc.dat","stu1.dat");
                    break;
            case '2':
                    clrscr();
                    box();
                    edit();
                    gotoxy(32,5);
                    cout<<"Students Detail";
                    fin.open("stu1.dat",ios::in);
                    fin.seekg(0,ios::end);
                    rec=fin.tellg()/sizeof(objStudent);
                    fin.seekg(0,ios::beg);
                    row=10;
                              //**PRINTING COURSE COMPLETED
STUDENT DETAILS**

                    for(i=1;i<=rec;i++)
                    {
                      fin.read((char*)&objStudent,sizeof(objStudent));
                      if(strcmp(objStudent.returnStatus(),"C_Comp")==0)
                      {
                        objStudent.showStudentDetail(row++);
                        fin1.open("course1.dat",ios::in);
                        fin1.seekg(0,ios::end);
                        rec1=fin1.tellg()/sizeof(objCourse);
                        fin1.seekg(0,ios::beg);
                        for(j=1;j<=rec1;j++)
                        {
                          fin1.read((char*)&objCourse,sizeof(objCourse));
                          if(objStudent.returnCid1()==objCourse.returnCid())
                          {
                              strcpy(x,objCourse.returnName());
                              cout<<strupr(x);
                          }
                        }
                        fin1.close();
                      }

                    }
                    fin.close();
                    getch();
                    break;
            case '3':
                    break;
           }
        }while(ch4!='3');
```

```
            break;
        case '4':
            do
            {
                clrscr();
                box();
                gotoxy(32,5);
                cout<<"Student Detail";
                gotoxy(29,9);
                cout<<"1. Course Left ";
                gotoxy(29,11);
                cout<<"2. Display CL List ";
                gotoxy(29,13);
                cout<<"3. Exit";
                gotoxy(29,18);
                cout<<" Enter your choice:- ";
                ch4=getche();
                switch(ch4)
                {
                  case '1':
                        clrscr();
                        box();
                        edit();
                        gotoxy(32,5);
                                //**PRINTING STUDENTS DETAILS ALONG
WITH COURSE NAME**
                        cout<<"Students Detail";
                        fin.open("stu1.dat",ios::in);
                        fin.seekg(0,ios::end);
                        rec=fin.tellg()/sizeof(objStudent);
                        fin.seekg(0,ios::beg);
                        row=10;
                        for(i=1;i<=rec;i++)
                        {
                           fin.read((char*)&objStudent,sizeof(objStudent));
                           objStudent.showStudentDetail(row++);
                           fin1.open("course1.dat",ios::in);
                           fin1.seekg(0,ios::end);
                           rec1=fin1.tellg()/sizeof(objCourse);
                           fin1.seekg(0,ios::beg);
                           for(j=1;j<=rec1;j++)
                           {
                             fin1.read((char*)&objCourse,sizeof(objCourse));
                             if(objStudent.returnCid1()==objCourse.returnCid())
                             {
                                strcpy(x,objCourse.returnName());
                                cout<<strupr(x);
                             }
                           }
                           fin1.close();
                        }
                        fin.close();
                        gotoxy(10,21);
                                //**TAKING ROLL NUMBER TO MARK HIS
```

```
COURSE LEFT**
                                        cout<<"Enter the rollno of student who left the course:-
";
                                        cin>>rollno;
                                        fin.open("stu1.dat",ios::in);
                                        fin1.open("cc.dat",ios::out);
                                        fin.seekg(0,ios::end);
                                        rec=fin.tellg()/sizeof(objStudent);
                                        fin.seekg(0,ios::beg);
                                             //**MARKING GIVEN ROLL NUMBER AS
COURSE LEFT**
                                        for(i=1;i<=rec;i++)
                                        {
                                          fin.read((char*)&objStudent,sizeof(objStudent));
                                          if(objStudent.returnRoll()==rollno)
                                          {
                                             objStudent.setStatus("C_Left");
                                             fin1.write((char*)&objStudent,sizeof(objStudent));

                                          }
                                          else
                                          {
                                             fin1.write((char*)&objStudent,sizeof(objStudent));
                                          }
                                        }
                                        fin.close();
                                        fin1.close();
                                        remove("stu1.dat");
                                        rename("cc.dat","stu1.dat");
                                        break;
                                case '2':
                                        clrscr();
                                        box();
                                        edit();
                                        gotoxy(32,5);
                                        cout<<"Students Detail";
                                        fin.open("stu1.dat",ios::in);
                                        fin.seekg(0,ios::end);
                                        rec=fin.tellg()/sizeof(objStudent);
                                        fin.seekg(0,ios::beg);
                                        row=10;
                                             //**PRINTING COURSE LEFT STUDENT
DETAILS**
                                        for(i=1;i<=rec;i++)
                                        {
                                         fin.read((char*)&objStudent,sizeof(objStudent));
                                         if(strcmp(objStudent.returnStatus(),"C_Left")==0)
                                         {
                                           objStudent.showStudentDetail(row++);
                                           fin1.open("course.dat",ios::in);
                                           fin1.seekg(0,ios::end);
                                           rec1=fin1.tellg()/sizeof(objCourse);
                                           fin1.seekg(0,ios::beg);
                                           for(j=1;j<=rec1;j++)
```

```
                            {
                             fin1.read((char*)&objCourse,sizeof(objCourse));
                             if(objStudent.returnCid1()==objCourse.returnCid())
                             {
                                 strcpy(x,objCourse.returnName());
                                 cout<<strupr(x);
                             }
                            }
                            fin1.close();
                           }
                          }
                          fin.close();
                          getch();
                          break;
               case '3':
                          break;
              }
        }while(ch4!='3');
        break;
case '5':
        clrscr();
        box();
        edit();
        gotoxy(32,5);
                          //**PRINTING ALL STUDENT DETAILS**
        cout<<"Students Detail";
        fin.open("stu1.dat",ios::in);
        fin.seekg(0,ios::end);
        rec=fin.tellg()/sizeof(objStudent);
        fin.seekg(0,ios::beg);
        row=10;
        for(i=1;i<=rec;i++)
        {
             fin.read((char*)&objStudent,sizeof(objStudent));
             objStudent.showStudentDetail(row++);
             fin1.open("course1.dat",ios::in);
             fin1.seekg(0,ios::end);
             rec1=fin1.tellg()/sizeof(objCourse);
             fin1.seekg(0,ios::beg);
             for(j=1;j<=rec1;j++)
             {
               fin1.read((char*)&objCourse,sizeof(objCourse));
               if(objStudent.returnCid1()==objCourse.returnCid())
               {
                    strcpy(x,objCourse.returnName());
                    cout<<strupr(x);
               }
             }
             fin1.close();
        }
        fin.close();
        getch();
        break;
case '6':
```

```
                      break;
              }
             }while(ch2!='6');
             break;
    // **INSIDE FEE DATABASE**
case '3':
            clrscr();
            box();
            edit();
            fee objFee;
            int rno=0,rcptno=0;
            float amt=0,tFee=0,bFee=0,pFee;
            char dt[20];
            gotoxy(12,10);
            cout<<"Reciept No.:- ";
            gotoxy(12,12);
            cout<<"Date:- ";
            gotoxy(12,14);
            cout<<"Roll No.:- ";
            gotoxy(12,16);
            cout<<"Name:- ";
            gotoxy(12,18);
            cout<<"Course:-  ";
            gotoxy(50,10);
            cout<<"Total Fee:- ";
            gotoxy(50,12);
            cout<<"Balance:- ";
            gotoxy(50,14);
            cout<<"Enter Amount:- ";
    //**CALLING currentRecieptNo FUNCTION TO GENERATE RECIEPT NO.
            rcptno=currentRecieptNo();
            gotoxy(32,10);
            cout<<rcptno;
            gotoxy(32,12);
            gets(dt);
            gotoxy(32,14);
            cin>>rno;
            tFee=courseFee(rno);   //**GETTING TOTAL FEE**
            gotoxy(70,10);
            cout<<tFee;
            pFee=totalFee(rno);    //**GETTING PAID FEE**
            bFee=tFee - pFee;      //**CALCULATING BALANCE FEE**
            gotoxy(70,12);
            cout<<bFee;
            if(bFee<=0)
            {
             gotoxy(65,14);
             cout<<"FEE COMPLETED";
             getch();
            }
            else
            {
             gotoxy(70,14);
             cin>>amt;
```

```
                                        }
                                        fin.open("fee1.dat",ios::app);
                                        objFee.getFeeDetail(rno,rcptno,dt,amt);
                                        fin.write((char*)&objFee,sizeof(objFee));
                                        fin.close();
                                        break;
                         case '4':
                                        break;
                     }
         }while(ch1!='4');
         end();
         clrscr();
      }
```