Institutional Practical Training Project Report

On

" Student Database Management System "

Submitted in the Partial fulfillment of the requirement for the Award of Degree of

Bachelor of Technology

in

COMPUTER SCIENCE & ENGINEERING

Batch
(2017-21)

Submitted to
HOD(CSE)

Submitted by
Shekhar Kashyap (1701367)
Shrish Saini (1701370)
Sanchay Khandelwal (1701355)
Sumit Kumar Jha (1701384)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Amritsar College of Engineering & Technology, Amritsar

(Autonomous college under UGC Act – 1956[2(f) and 12(B)])

ACKNOWLEDGEMENTS

This is a humble effort to express my sincere gratitude towards those who have guided and helped me to complete this project

A project is major milestone during the study period of a student. As such this project was a challenge to me and was an opportunity to prove my caliber. I am highly grateful and obliged to each and everyone making me help out of problems being faced by me.

It would not have been possible to see through the undertaken project without the guidance of **Er. Shivani, Er. Harpinder Singh and Er. Parambir Singh** . It was purely on the basis of their experience and knowledge that we able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least I am very thankful to our Head of Department **Er. Amarpreet Singh** and all Members of Computer Science Deptt. who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them.

DECLARATION

I Name hereby declare that the project work entitled **"Student Database Management System"** is an authentic record of my own work carried out as requirements of Institutional Training project for the award of degree of B.Tech(CSE), **Amritsar College of Engg. And Technology, Amritsar,** under the guidance of **Er. Shivani, Er. Harpinder Singh and Er. Parambir Singh**

(Signature of students)

Shekhar Kashyap (1701367)
Shrish Saini (1701370)
Sanchay Khandelwal (1701355)
Sumit Kumar Jha (1701384)

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

Faculty Coordinator

Er. Shivani Sharma (Assistance Professor – CSE Department)

Er. Harpinder Singh (Assistant Professor – CSE Department)

Er. Parambir Singh (Assistant Professor – CSE Department)

INDEX PAGE

Introduction to C++

The Origins of C++

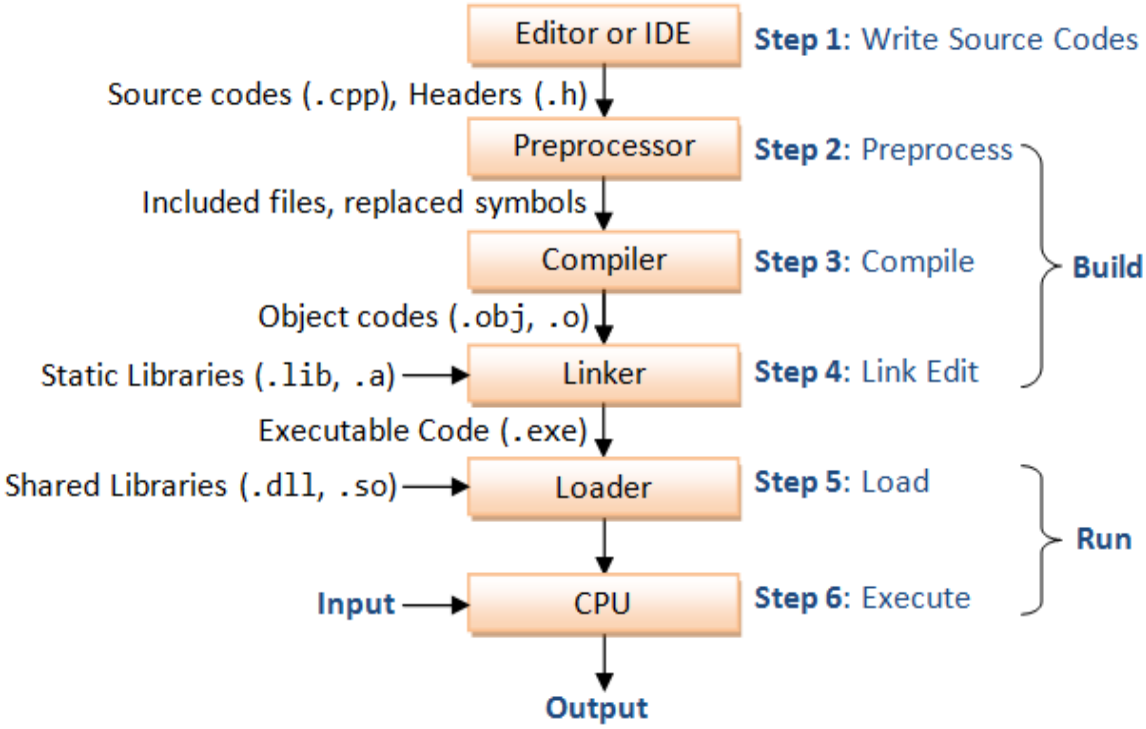
C++ was developed by Bjarne Stroustrup of AT&T Bell Laboratories in the early 1980's, and is based on the C language. The "++" is a syntactic construct used in C (to increment a variable), and C++ is intended as an incremental improvement of C. Most of C is a subset of C++, so that most C programs can be compiled (i.e. converted into a series of low-level instructions that the computer can execute directly) using a C++ compiler

The Programming Environment

We need several pieces of software:

- An editor with which to write and modify the C++ program components or source code,
- A compiler with which to convert the source code into machine instructions which can be executed by the computer directly,
- A linking program with which to link the compiled program components with each other and with a selection of routines from existing libraries of computer code, in order to form the complete machine-executable object program,
- A debugger to help diagnose problems, either in compiling programs in the first place, or if the object program runs but gives unintended results

Program Execution Stages



The Features of C++

C++ is the multi paradigm, compile, free form , general purpose, statistically typed programming language. This is known as middle level language as it comprises of low level and high level language features.

And there are some other things and advantages of this language over the C. The first commercial implementation of the C++ was released in 1985 and before that the name of language was changed to "C++". And some new features were added to the language and The main features of the C++ are

- Classes
- Inheritance
- Data abstraction and encapsulation
- Polymorphism
- Dynamic Binding

- Message Passing

1) **Classes:** By using classes we can create user defined data types. In other words the class is the collection of set of data and code. The class allows us to do some things which are polymorphism, inheritance, abstraction, encapsulation which are our next features. The objects are the instances of classes.

The syntax for class is :
Class <class-name>
{
//Body of class;
};

2) **Inheritance:** Inheritance allows one data type to acquire properties of other data types. Inheritance from a base class may be declared as public, protected, or private. If the access specifier is omitted, a "class" inherits privately, while a "struct" inherits publicly. This provides the idea of reusability that means we can add the new features to an existing class without modifying it.

3) **Data Abstraction and Encapsulation:** Encapsulation means hiding of data from the data structures or in other words wrapping up of data in single entity is known as Encapsulation. In this the data is not accessible to outside world and only the functions are allowed to access it. When we want to write the class in which we don't have the knowledge about the arguments used to instantiate it then we can use templates in C++. Abstraction can be defined as the act of representing essential features without including background details.

4) **Polymorphism:** it means that the one interface can be used for many implementation so that object can behave differently for each implementation. The different types of polymorphism are static (Compile time) and dynamic (Run time).

5) **Dynamic Binding:** It means that the linking of a procedure call to code to be executed in response to the call. A function call associated with a polymorphic reference depends on the dynamic type that reference. And at run-time the code matching the object under current reference will be called.

6) **Message Passing:** An object oriented program consists of the set of objects that communicate with each other. objects communicate with one another by sending and receiving information much the same

way as people pass messages to one another. The concept of message passing makes it easier to direct model or simulate their real world counterparts.

OBJECTIVES OF THE PROJECT

The main objective of the proposed system is that any modification can be made by just the touch of a button instead of going through directory and keep on turning pages.

Proposed system will contain:

1. Information of all the Student's i.e. personnel & academic details..
2. It keeps track of all the students Details.
3. It keeps track of all the Course's Details at university & college levels.
4. Dynamic generation of students placement graph.
5. Preview slideshow of courses available at college.
6. Capable of Generating Dynamic reports such as student details, academic details, duplicate admit card/result card printing.
7. Contact Information about all the Enginnering students..

Source Code :

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<dos.h>

fstream f1,f2,f3,f4,f6,f7;
class student
{
public:
char name[50],fname[30],address[500],department[30];
long int rn, reg,pno,sem,l;
void input()
{
cout<<"Enter Student Name:";
gets(name);
cout<<"Enter Father Name:";
gets(fname);
cout<<"Enter Student Roll No:";
cin>>rn;
cout<<"Enter Address:";
gets(address);
cout<<"Enter Registeration No:";
cin>>reg;
cout<<"Enter Department:";
gets(department);
cout<<"Enter Phone No:";
cin>>pno;
cout<<"Enter Semester:";
cin>>sem;
}
void display()
{
cout<<"NAME              :";
puts(name);
cout<<"FATHER-NAME       :";
puts(fname);
cout<<"ROLL-NO           :"<<rn;
cout<<"\nADDRESS           :";
puts(address);
cout<<"REGISTERATION-No   :"<<reg;
cout<<"\nDEPARTMENT        :";
puts(department);
```

```
cout<<"PHONE-NO          :"<<pno;
cout<<"\nSEMESTER          :"<<sem;
}
};
void main()
{
student A;
long int n,rn1,m,i,ch,s,g,j,y;
char d,u[5]="abc",r[5],p[5]="acet",p1[5];
textbackground(WHITE);
textcolor(RED);
clrscr();
cout<<"##################################################################################
#########\n";
cout<<"\n\n";
cout<<"\t  ///////////    ///////////   ////////////    /////////////// \n";
cout<<"\t  //      //    //           //                //        \n";
cout<<"\t  //      //    //           //                //       \n";
cout<<"\t  //      //    //           //                //       \n";
cout<<"\t  //      //    //           ////////////       //       \n";
cout<<"\t  /////////     //           //                //       \n";
cout<<"\t  //      //    //           //                //       \n";
cout<<"\t  //      //    //           //                //       \n";
cout<<"\t  //      //    ///////////   ////////////       //       \n";
cout<<"\n\n\n";
cout<<"\t\t    ////////////////////////////////// \n";
cout<<"\t\t    //  STUDENT DATABASE MANAGEMENT SYSTEM   //\n";
cout<<"\t\t    ////////////////////////////////// \n";
cout<<"\n";
cout<<"\t\t       Press Enter to Continue.....\n";
cout<<"\n##################################################################################
###########\n";
getch();
mno:
textbackground(WHITE);
textcolor(BLUE);
clrscr();
cout<<endl<<"\t\t\t====================\n\t\t\tDASHBOARD
LOGIN"<<endl<<"\t\t\t===================="<<endl<<endl;
cout<<endl<<endl<<endl<<"\t\tUSERNAME::";
gets(r);
cout<<endl<<endl<<"\t\tPASSWORD::" ;
//j=p;
i=0;
while(i<4)
{
p1[i]=getch();
cout<<"*";
i++;}
```

```
int v=0,q=0;
for(i=0;i<4;i++)
{ if(p1[i]==p[i])
{v++;}}
for(i=0;i<3;i++)
{ if(r[i]==u[i])
{q++;}}

if(v==4 && q==3)
{ cout<<"\n";
goto sd;
}
else
{
cout<<endl<<endl<<endl<<"\t\t\t\t::!!!Invalid Password/Username";
cout<<"\n ..::Try again?\n\n\t[1] Yes\t\t[0] No\n\t";
cin>>s;
if(s==1)
{
goto mno;
}
else
{
exit(1);
}
}
sd:
cout<<endl<<endl<<endl<<"\t\t\t\t..::SUCCESSFULLY LOGIN.....";
delay(1000);
 textbackground(WHITE);
 textcolor(RED);
clrscr();
cout<<"\n\n\n\n\n\n\n\n\n\n\n\t\t**** Welcome to Student Record Management System ****";
cout<<endl<<endl<<endl<<"\tLoading:";
for(i=0; i<20; i++)
{      cout<<"...";
       delay(50);
           }
abc:
 textbackground(WHITE);
 textcolor(RED);
clrscr();
cout<<"\n\t **** Welcome to Student Record Management System ****";
cout<<"\n\n\n\t\t\tMAIN  MENU\n\t\t====================\n\t\t[1]  Add  Record\n\t\t[2]
Search Record\n\t\t[3] View All Records\n\t\t[4] Update Record\n\t\t[5] Delete Record\n\t\t[6]
Exit\n\t\t=================\n\t\t";
cout<<"Enter Your Choice:";
cin>>n;
switch(n)
```

```
{
case 777:
abd:
textbackground(RED);
textcolor(WHITE);
cout<<endl<<"\t\t::Enter Your Choice:";
cout<<endl<<endl<<endl<<endl<<"     [0] Main Menu                [5] Exit";
cout<<endl;
cin>>ch;
clrscr();
if(ch==0)
{
goto abc;
}
else
{
 textbackground(WHITE);
 textcolor(RED);
cout<<"\n################################################################################
###########\n";
cout<<endl<<endl<<endl<<"\t\tAre You Sure You Want To Exit(y/n):     ";
cin>>d;
//cout<<"\n\n\n\n\n\n\n#############################################################################
#####################\n";
if(d=='y')
{
cout<<"\n\n\n\n\t\t**************THANK YOU**********************\n\n\n\n\n";
for(i=0; i<70; i++)
{
cout<<"#";
      delay(40);
          }
exit(1);
}}
goto abc;
break;
case 1:
textbackground(WHITE);
textcolor(RED);
clrscr();
cout<<"\n\t\t================================\n\t\t\tADD
RECORD\n\t\t================================\n\n";
cout<<"Enter No of Records To Add::::";
cin>>m;
textbackground(WHITE);
textcolor(RED);
clrscr();
cout<<"\n\t\t================================\n\t\t\tADD
RECORD\n\t\t================================\n\n";
```

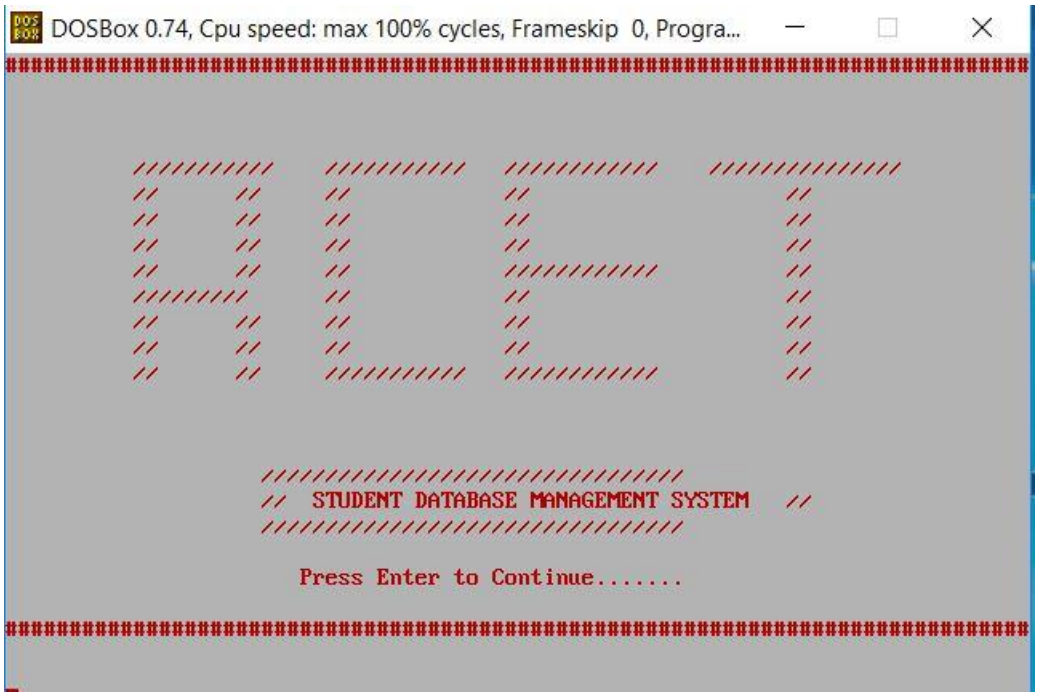
```
f4.open("def12.dat",ios::app|ios::binary);
for(i=0;i<m;i++)
{
A.input();
f4.write((char *)&A,sizeof(A));
cout<<endl<<endl<<"\t\tData Saved Successfully"<<endl<<endl;
}
f4.close();
goto abd;
break;
case 2:
cde:
textbackground(WHITE);
textcolor(RED);
clrscr();
f1.open("def12.dat",ios::in|ios::binary);
cout<<"\n\n\t..::RECORD SEARCH\n\t==========================\n\t..::Roll No of
Student to search: ";
cin>>rn1;
clrscr();
f1.seekg(0);
while(f1.read((char *)&A,sizeof(A)))
{
if(rn1==A.rn)
{
textbackground(WHITE);
textcolor(RED);
cout<<"\n\n..::Search result for\t<<"
<<rn1<<">>\n================================================\n";
A.display();
getch();
f1.close();
goto abd;
getch();
exit(1);
}}
f1.close();
textbackground(WHITE);
textcolor(RED);
cout<<"\n\n..::Search result
for\t"<<rn1<<"\n================================================\n";
cout<<"\n..::No match found!";
cout<<"\n ..::Try again?\n\n\t[1] Yes\t\t[0] No\n\t";
cin>>s;
if(s==1)
{
goto cde;
}
else
```

```
{
f1.close();
goto abd;
}
break;
case 3:
clrscr();
f6.open("def12.dat",ios::in|ios::binary);
cout<<"\n\t\t================================\n\t\t\tLIST                    OF
RECORDS\n\t\t================================";
//f1.seekg(0);
while(f6.read((char *)&A,sizeof(A)))
{
textbackground(WHITE);
textcolor(RED);
cout<<"\n============================================================
====\n\n";
A.display();
getch();
}
f6.close();
goto abd;
break;
case 4:
wer:
textbackground(WHITE);
textcolor(RED);
clrscr();
f7.open("def12.dat",ios::in|ios::out|ios::binary);
cout<<"\n\n\t..::RECORD  UPDATE\n\t==========================\n\t..::Roll  No  of
Student to Update Record: ";
cin>>rn1;
clrscr();
while(f7.read((char *)&A,sizeof(A)))
{
f7.seekg(0,ios::cur);
if(rn1==A.rn)
{
textbackground(WHITE);
textcolor(RED);
int pos=f7.tellg()-sizeof(A);
f7.seekp(pos,ios::beg);
cout<<endl<<"Write To Update Data:"<<endl<<endl<<endl;
A.input();
f7.write((char *)&A,sizeof(A));
cout<<endl<<endl<<endl<<"Record Updated Successfully"<<endl<<endl;
f7.close();
goto abd;
getch();
```

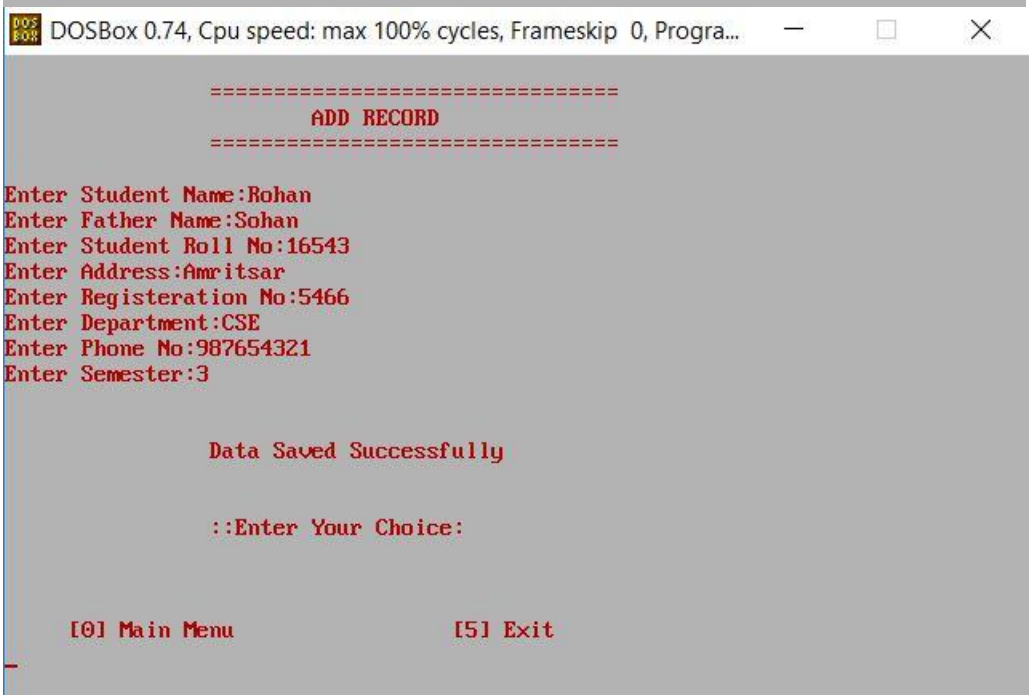
```
exit(1);
}}
textbackground(WHITE);
textcolor(RED);
cout<<"\n\n..::Search result for
\n================================================\n";
cout<<"\n..::No match found!";
cout<<"\n ..::Try again?\n\n\t[1] Yes\t\t[0] No\n\t";
cin>>s;
if(s==1)
{
goto wer;
}
else
{
f7.close();
goto abd;
}
break;
case 5:
//int rn1;
textbackground(WHITE);
textcolor(RED);
clrscr();
//fstream f2;
f3.open("def12.dat",ios::in|ios::binary);
f2.open("temp125.dat",ios::out|ios::binary|ios::ate|ios::in);
cout<<"\n\n\t..::RECORD DELETE\n\t==========================\n\t..::Roll No of
Student to Delete Record: ";
cin>>rn1;
clrscr();
while(f3.read((char*)&A,sizeof(A)))
{
if(rn1!=A.rn)
{
textbackground(WHITE);
textcolor(RED);
//ofstream f1;
//f1.open("record3.txt");
//A.display();
f2.write((char*)&A,sizeof(A));
}}
f3.close();
f2.close();
remove("def12.dat");
rename("temp125.dat","def12.dat");
cout<<endl<<"Record Deleted Successfully"<<endl<<endl;
goto abd;
break;
```

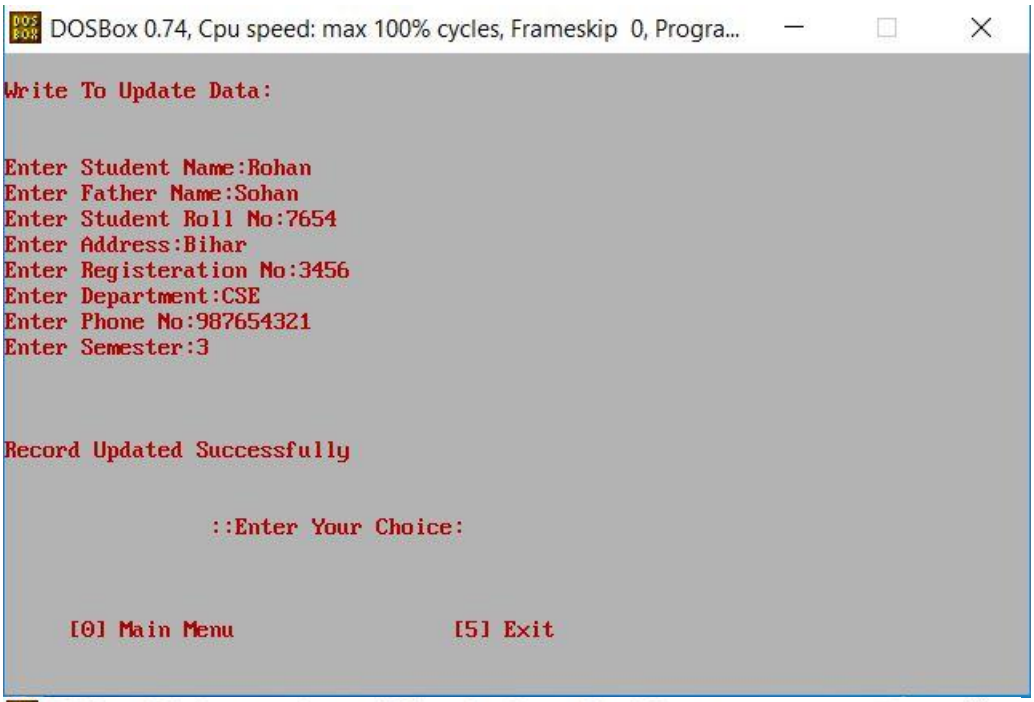
```
case 6:
cout<<endl<<endl<<endl<<"\t\tAre You Sure You Want To Exit(y/n):      ";
cin>>d;
if(d=='y')
{
exit(1);
}
else
{
clrscr();
goto abc;
}
break;
}
getch();
}
```

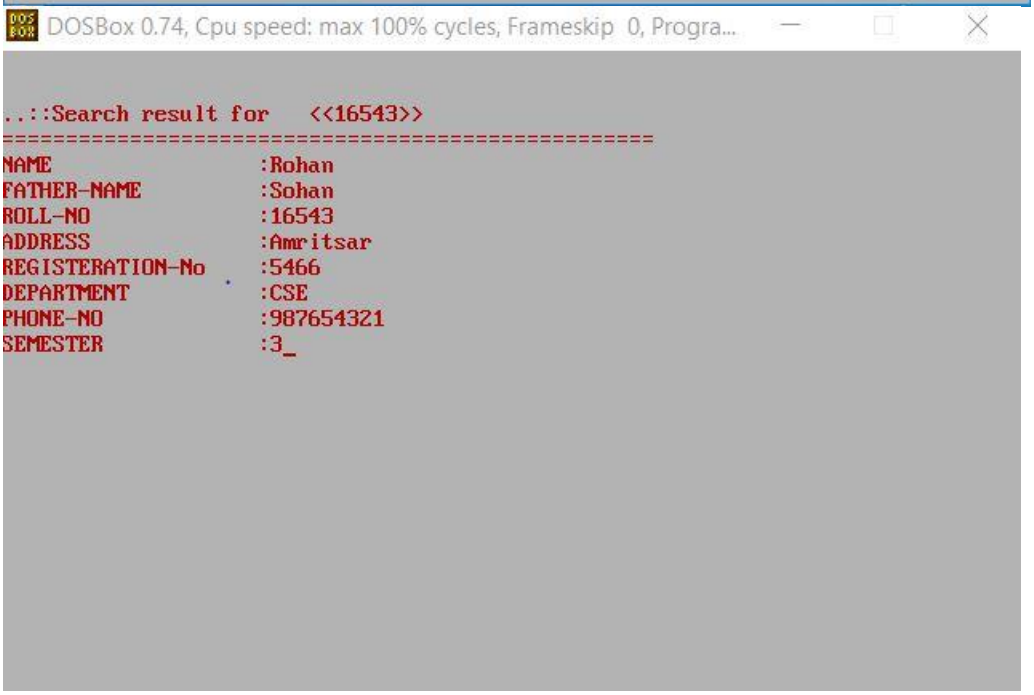
OUTPUT:

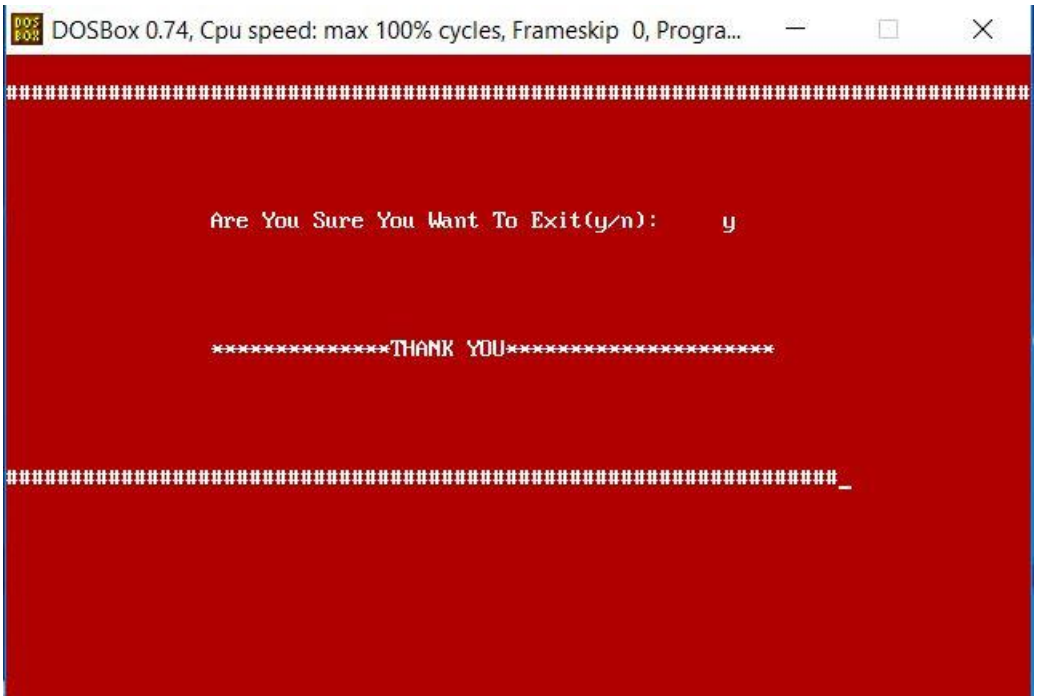



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
**** Welcome to Student Record Management System ****
MAIN MENU
[1] Add Record
[2] Search Record
[3] View All Records
[4] Update Record
[5] Delete Record
[6] Exit
Enter Your Choice:_


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
ADD RECORD
Enter Student Name:Rohan
Enter Father Name:Sohan
Enter Student Roll No:16543
Enter Address:Amritsar
Enter Registeration No:5466
Enter Department:CSE
Enter Phone No:987654321
Enter Semester:3
Data Saved Successfully
::Enter Your Choice:
[0] Main Menu
[5] Exit


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Write To Update Data:
Enter Student Name:Rohan
Enter Father Name:Sohan
Enter Student Roll No:7654
Enter Address:Bihar
Enter Registeration No:3456
Enter Department:CSE
Enter Phone No:987654321
Enter Semester:3
Record Updated Successfully
::Enter Your Choice:
[0] Main Menu
[5] Exit


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
..::Search result for <<16543>>
NAME :Rohan
FATHER-NAME :Sohan
ROLL-NO :16543
ADDRESS :Amritsar
REGISTERATION-No :5466
DEPARTMENT :CSE
PHONE-NO :987654321
SEMESTER :3_


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Are You Sure You Want To Exit(y/n): y
**********THANK YOU**********

References

1. **Books**

   - C++ Programming by Anshuman Sharma
   - Programming in C++ by Robert Lafore
   - C++ Concepts by R S Salaria

2. **Web URLs**

   - https://ameyasworld.wordpress.com/2010/11/27/features-of-c/
   - www.ntu.edu.sg/home/ehchua/programming/cpp/cp1_**Basics**.html
   - www.tutorialspoint.com/cplusplus/cpp_**files**_streams.htm
   - www.codinpsycho.blogspot.com/2011/11/**file**-**handling**-in-c++**basics**.html
   - www.csegeek.com/csegeek/view/tutorials/cpp_lang/cpp_**file**.php