**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

****

**COURSE PROJECT REPORT ON:**

**“HOSTEL MANAGEMENT SYSTEM**”

**[Project report submitted on partial fulfilment of the requirements of 4th semester COMPUTER SCIENCE AND ENGINEERING]**

**OBJECT ORIENTED PROGRAMMING CONCEPT**

**PROJECT REPORT BY:**

MANORANJAN.M 1NT12CS081

SACHIN 1NT12CS134

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**[2013-2014]**

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(An Autonomous Institution)  
Approved and Accredited by AICTE,   
Affiliated to Visvesvaraya Technological University  
**Yelahanka, Bangalore - 560 064.**

****

**CERTIFICATE**

**This is to certify that the project report entitled**

**“**HOSTEL MANAGEMENT SYSTEM**”**

Is an authentic record of the project work carried out by

MANORANJAN.M 1NT12CS081

SACHIN 1NT12CS134

In partial fulfilment of the requirements for the award of completion of **“HOSTEL MANAGEMENT SYSTEM** ” project under the guidance and supervision of concerned lecturer during the year 2014.

**SIGNATURE OF HOD SIGNATURE OF EXAMINER**

.................................. ................................... .................................. ...................................

(Dr. Nalini N) (Asst.Prof. Prathiba Ballal)

**Department of Computer Science & Engineering**

**[2013-14]**

**ACKNOWLEDGEMENT**

We consider it our privilege to express our gratitude and respect to all those who guided us in the completion of this project.

We are grateful to **Dr. NALINI N,** Head of the Department, Computer Science and Engineering for providing support and encouragement.

we express our sincere thanks and gratitude to **Mrs. PRATHIBHA BALLAL**,Asst.Prof, Computer Science and Engineering for her able guidance and support through the project.

We would also like to express thanks to our friends and parents for providing inspiration to complete project.

We would also like to thank **Dr.H.C.NAGARAJ**, Principal, NMIT, Bangalore for encouraging course project in our ciriculum.

**ABSTRACT**

The program “HOSTEL MANAGEMENT SYSTEM” is using some major concepts of oop using C++.This program is simple and object oriented data base. In this program oop employee’s the bottoms up approach.

Oop treats data as critical element in the program development and does not allow it to flow freely around the system. A class is a way to bind the data and its associated functions together and it allows data and functions to be hidden if necessary from external use.

A class is an abstract data type that can be treated like any other built in data type. A problem is considered as a collection of a number of entities called objects, objects are instances of classes, and insulation of data by direct access by the program is called Data hiding. Data abstraction refers to inserting together essential features without including background details.

C++ is a statically typed, freeform, mulch paradigm complied, general purpose programming language. It is regarded as an intermediate level language as it comprises a combination of both high level language features developed by Bjarne Stroustrup starting in 1979 at Bell labs, c++ was originally named c with classes, adding object oriented features such as classes and other enhancements to the c programming language the language was renamed as c++ in 1983 as pun involving the increment operator.

C++ is also used for hardware design, where the design is initially described in c++ then analyzed architecturally constructed and scheduled to create a register transfer level hardware description via high level synthesis.

**TABLE OF CONTENTS**

CONTENTS PAGE NO.

01. INTRODUCTION. 07

02. LITERATURE SURVEY 08-12

03. DESIGN OF THE SYSTEM. 13

04. PROGRAM CODE. 14-23

05. SNAPSHOTS. 24-30

06. CONCLUSION. 31

07. BIBLIOGRAPHY. 31

**INTRODUCTION**

* The project is all about managing a hostel’s and mess’s data.
* The main user is the hostel’s supervisor, who can store and access the details.
* By the help of this program any user would be able to do the following things :
* Manage entries of staff and students.
* Keep the record of a student’s mess bill and keep a check on the paid and unpaid mess bills..
* It is even possible to see a student’s record if He/She had already left hostel.
* We can access any student’s complete details.
* The project provides a friendly interface to the user so that it is easy for any user to use this program to its full extent.
* The project makes a use of following different features of OOP :
  + - Inheritance.
    - Classes and Objects.
    - Function Overloading.
    - Constructors.
    - Friend Function.
    - Operator Overloading.
    - File Handling.

**Literature survey**

**C++ Class concept:**

A class is a way to bind the data and its associated

functions together. It allows data and functions to be hidden, if necessary from external use.

A class is an abstract data type that can be treated like any other built-in data type.

Generally, a class specification has two parts: Class declaration and class function definations.

Class class\_name

{

Private:variable declarations;

Function declaration;

};

The class members that have been declared as private can be accessed only within the class. Public members can be accessed outside the class also. Keyword private is optional. By default all members are private.

The variables declared inside the class are known as Data members and functions are known as member functions.

Only the member function can have access to private Data members and private functions. The public members can be accessed outside the class.

An object is a variable type class.

Member functions can be accessed as

Object\_name.function\_name(actual arguments);

Member function definition is as follows

Return\_type class\_name

{

Function body

}

**Inheritance:**

The mechanism of deriving a new class from an old class is called Inheritance. The derived class inherits some or all properties from the class. A derived class with only one base class is called Single Inheritance and one with many classes is called Multiple Inheritance.

The properties of one class can be inherited by more than one class. This is known as Hierarchical Inheritance.

Syntax of Inheritance

Class derived\_class\_name:visibility\_modebase\_class\_name

{

//members of derived class

};

**Constructors:**

A constructors is a special member function whose task is to minimize the objects of the class. Its name is same as class name.

A constructor is created whenever an object of its associated class is created.

A constructor is defined as follows:

Class integer

{

Int m,n;

Public:integer(); //constructor declared

……………….

};integer::integer() //constructor defined

{

M=0,n=0;

}

When a class contains a constructor,an object created by the class will be initialized automatically.

A constructor that accepts no parameters is called default constructor.

Example:

Class integer

{

Intm,n;

Public:integer()

{

M=0,n=0;

}

…………

};

**Friend Function:**

C++ allows a common function to be made friendly with classes,thereby allowing the function to have access to the private data of these classes.

To make an outside function friendly to a class,we have to simply declare this function as a friend function of the class

Example:

Class ABC

{…………

Public:………..

Friend void xyz(void)://declaration

}

The definition of the friend function does not either use the keyword friend or scope resolution operator::.

A function can be declared friend to any number of classes.

A friend function is invoked like a normal function. It cannot be called using object of the class.

**Operator Overloading:**

C++ has the ability to provide special meaning to an operator. This mechanism of giving special meaning to an operator is known as operator overloading. Operator overloading is done with the help of an operator function. The general form of the function is as follows:

Return\_typeclassname::operator op(arg list)

{ Function body//task defined for the operator

}

Where op is the operator being overloaded.

Operator function must be either be a member function or a friend function. If it is a friend function then unary operator will have one operand and binary operator will have two operand.

If the function is a binary function then it will have one operand and unary will have no operand.

Example:

{

Class COMPLEX

{

Int x;

Float y;

Public:

……………………

COMPLEX operator+(COMPLEX)

……………………

……………………

};

COMPLEX COMPLEX::operator+(COMPLEX c)

{

COMPLEX temp;

Temp.x=x+c.x;

Temp.y=y+temp.y;

Return temp;

}

Overload operator+ can be invoked as c3=c1+c2; where c1,c2,c3 are COMPLEX variables. Here c1 is the invoking object and c2 plays the role of the argument to be sent to the operator function.

Syntax for overloading”<<”:

Ostream& operator<<(ostream& out,class name)

{

//body of the function

Return out

}

**File Handling:**

ofstream: Stream class to write on files

ifstream: Stream class to read from files

fstream: Stream class to both read and write from/to files.

These classes are derived directly or indirectly from the classes istream and ostream. We have seen objects whose types were these classes:cin is an object of class istream cout is an object of class ostream. Therefore, we have already been using classes that are related to our file streams. And in fact, we can use our file streams the same way we are already used to use cin and cout, with the only difference that we have to associate these streams with physical files.

**DESIGN OF THE SYSTEM**

LEVEL 1:

SUPERVISOR

INHERITANCE

STAFF

LEVEL 2:

STORED IN STAFF FILE

STORED IN STUDENT FILE

STUDENT

**PROGRAM CODE**

IMPLEMENTATION:

#include<iostream>

#include<fstream>

#include<string.h>

using namespace std;

class supervisor

{

public: char name[20];

char add[20];

char tellno[20];

char code[20];

supervisor(){}

supervisor(char \*n,char \*d,char \*a,char \*t)

{

strcpy(name,n);

strcpy(code,d);

strcpy(add,a);

strcpy(tellno,t);

}

void get\_supr();

void display();

};

void supervisor::get\_supr()

{

cout<<"Enter the Code:"<<endl;

cin>>code;

cout<<"Enter Name:"<<endl;

cin>>name;

cout<<"Enter Address:"<<endl;

cin>>add;

cout<<"Enter Tellno:"<<endl;

cin>>tellno;

}

void supervisor::display()

{

cout<<"code:"<<code<<"\n"<<"name:"<<name<<"\n"<<"address:"<<add<<"\n"<<"tellno:"<<tellno<<endl;

}

class staff :public supervisor

{

public: char work[20];

char quali[20];

staff(){}

staff(char \*w,char \*q)

{

strcpy(work,w);

strcpy(quali,q);

}

friend istream &operator>>(istream&,staff&);

friend ostream &operator<<(ostream&,staff);

};

istream &operator>>(istream&stream,staff & p)

{

p.get\_supr();

cout<<"Enter working field"<<endl;

stream>>p.work;

cout<<"Enter qualification"<<endl;

stream>>p.quali;

return stream;

}

ostream &operator<<(ostream&stream,staff p)

{

stream<<endl;

stream<<"Staff code is:"<<p.code<<"\n"<<"Name of the staff:"<<p.name<<"\n"<<"Address of the staff:"<<p.add<<"\n"<<"Telephone no:"<<p.tellno<<"\n"<<"Working field of staff:"<<p.work<<"\n"<<"Qualification:"<<p.quali<<endl;

stream<<endl;

return stream;

}

class student:public supervisor

{

public: char sem[20];

char roomno[20];

float bill;

student(){}

student(char \*s,char \*r)

{

strcpy(sem,s);

strcpy(roomno,r);

}

friend istream &operator>>(istream&,student&);

friend ostream &operator<<(ostream&,student);

float messbill(char);

};

float student::messbill(char code)

{

int \*prsnt;

float \*per,\*paid;

static float bal;

prsnt=new int;

per=new float;

paid=new float;

if(code==0)

{

cout<<"\n Messbill calculation not possible\n";

}

else

{

cout<<"\n Enter the no of days present:";

cin>>\*prsnt;

cout<<"\n Enter per day cost:";

cin>>\*per;

}

if(code!=0)

{

cout<<"Balance remaining:\n";

bal=((\*prsnt)\*(\*per))-(\*paid);

cout<<bal<<endl;

cout<<"\n Enter amount paid\n";

cin>>\*paid;

bal=((\*prsnt)\*(\*per))-(\*paid);

}

delete per;

delete prsnt;

delete paid;

return (bal);

}

istream &operator>>(istream&stream,student &p)

{

p.get\_supr();

cout<<"Enter semister of the student:"<<endl;

stream>>p.sem;

cout<<"Enter room no of student:"<<endl;

stream>>p.roomno;

return stream;

}

ostream &operator<<(ostream&stream,student p)

{

stream<<endl;

stream<<"Student code is: "<<p.code<<"\n"<<"Name of the student: "<<p.name<<"\n"<<"Address of the student: "<<p.add<<"\n"<<"Telephone no: "<<p.tellno<<"\n"<<"Semister of student: "<<p.sem<<"\n"<<"room no of student: "<<p.roomno<<endl;

stream<<endl;

return stream;

}

int main()

{

int i;

supervisor S;

student o[20];

staff s[20];

float bill;

int choice;

char test;

ofstream out("staff",ios::out|ios::app);

ifstream in("staff");

if(!in||!out)

{

cout<<"file not found\n";

return(0);

}

ofstream outst("Student",ios::out|ios::app);

ifstream inst("Student");

if(!inst||!outst)

{

cout<<"file not found\n";

return(0);

}

ofstream out("file",ios::out|ios::app);

ifstream in("file");

if(!in||!out)

{

cout<<"file not found\n";

return(0);

}

start:

cout<<"\n";

cout<<"\n\n!!!\*#\* PROJECT ON HOSTEL MANAGEMENT\*#\*!!!\n\n";

cout<<"\n";

cout<<"\n HOSTEL DETAILS \n";

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<"\n";

cout<<"choose category from the below menu\n";

cout<<"\n";

cout<<"1.SUPERVISOR\n";

cout<<"2.Exit\n";

cout<<"\n";

for(;;)

{

cout<<"Enter your choice\n";

cin>>choice;

switch(choice)

{

case 1:cout<<"\n";

cout<<"1.WORKING STAFF\n";

cout<<"2.STUDENT REPORT\n";

cout<<"3.EXIT\n";

cout<<"\n";

for(;;)

{

cout<<"Enter your choice\n";

cin>>choice;

switch(choice)

{

case 1: cout<<"\nSelect your choice from the below sub-menu\n";

cout<<"\n";

cout<<"1.ENTER THE STAFF RECORD\n";

cout<<"2.DISPLAY STAFF DATA\n";

cout<<"3.SKIP TO MENU\n";

for(;;)

{

cout<<"Your option?:\n";

cin>>choice;

switch(choice)

{

case 1: cout<<"ENTER NUMBER OF STAFF\n";

int n;

cin>>n;

for(i=0;i<n;i++)

{

in.read((char \*)&s,sizeof(class staff));

cin>>s[i];

out<<s[i];

in.close();

}

break;

case 2: out.write((char \*)&s,sizeof(class staff));

out.close();

char c;

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

while(in.get(c))

cout<<c;

in.clear();

break;

cout<<endl;

cout<<"\nAre you interested in feeding further entries under staff section?:\n";

cout<<"Please enter 'Y/y'for Yes and 'N/n' for No:\n";

cin>>test;

if(test=='y'||test=='Y')

continue;

else goto out1;

out1: break;

case 3: goto start;

}

}

case 2: cout<<"\nSelect your choice from the below sub-menu\n";

cout<<"\n";

cout<<"1.ENTER THE STUDENT RECORD\n";

cout<<"2.DISPLAY STUDENT DETAILS\n";

cout<<"3.bill\n";

cout<<"4.SKIP TO MENU\n";

for(;;)

{

cout<<"Your option?:\n";

cin>>choice;

switch(choice)

{

case 1: cout<<"ENTER NUMBER OF STUDENTS\n";

int n;

cin>>n;

for(i=0;i<n;i++)

{

inst.read((char \*)&o,sizeof(class student));

inst.close();

cin>>o[i];

outst<<o[i];

}

cout<<endl;

cout<<"\nAre you interested in feeding further entries under student section?:\n";

cout<<"Please enter 'Y/y'for Yes and 'N/n' for No:\n";

cin>>test;

if(test=='y'||test=='Y')

continue;

else goto out2;

out2: break;

case 2: outst.write((char \*)&o,sizeof(class student));

outst.close();

char c;

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

while(inst.get(c))

cout<<c;

inst.clear();

break;

case 3: char code;

cout<<"\n Still Balance remaining:"<<o[i].messbill(code)<<endl;

break;

case 4: goto start;

}

}

default: in.close();

out.close();

break;

}

case 5: goto end;

}

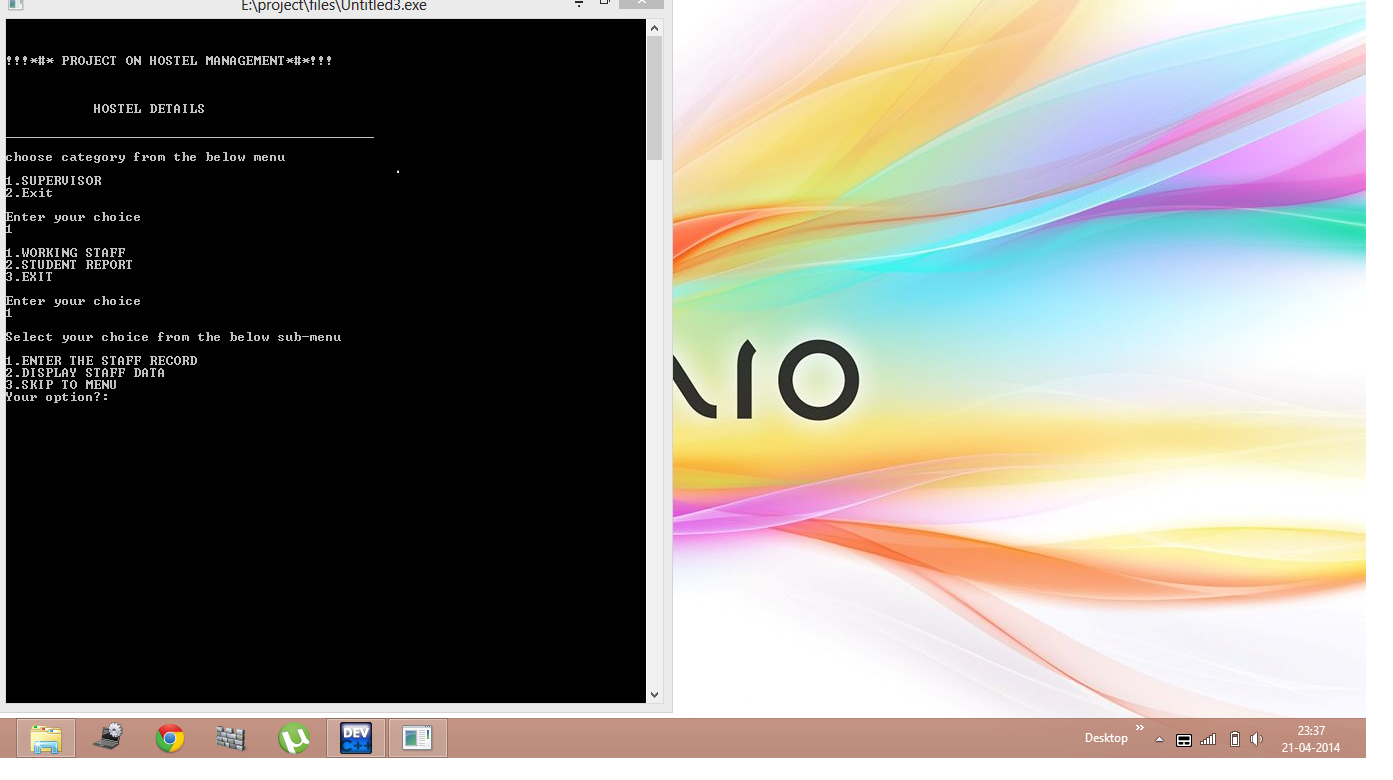
}

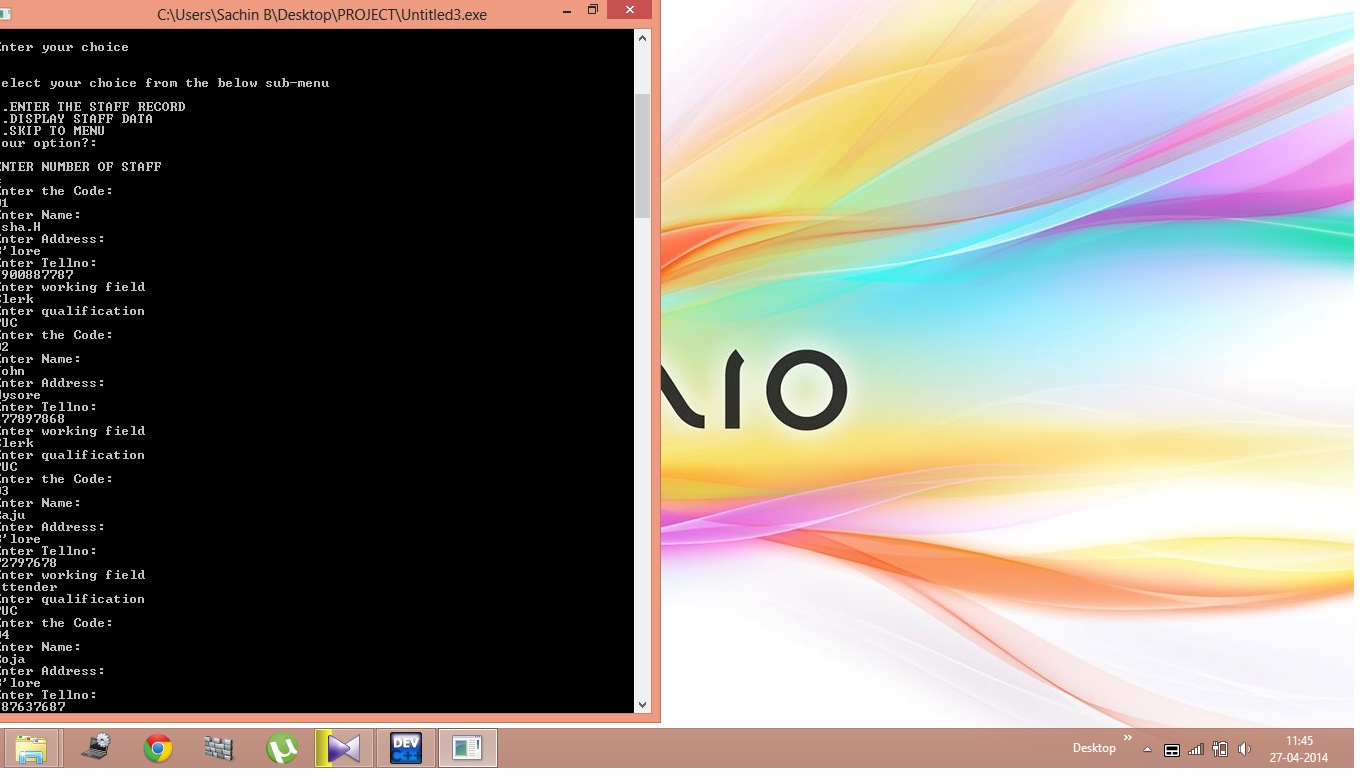
end: return(0);

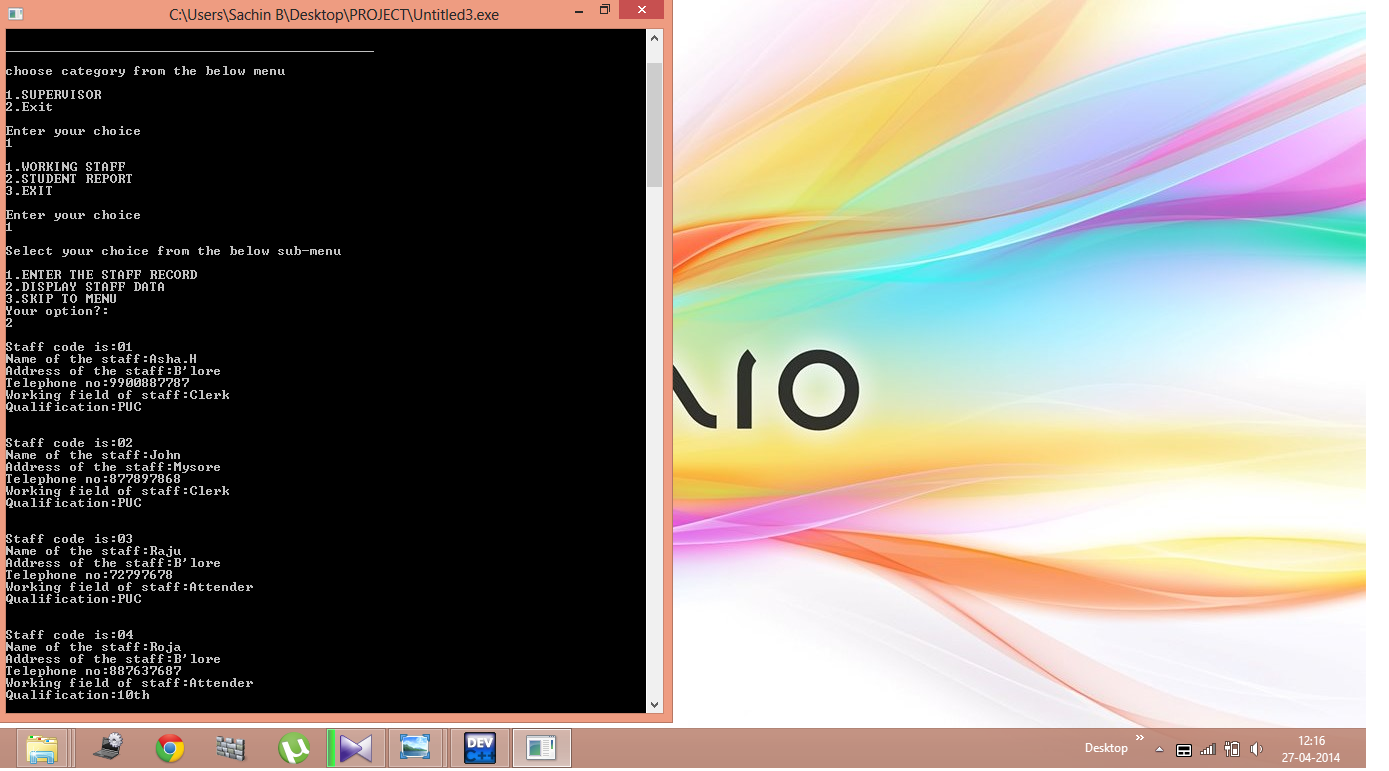
}

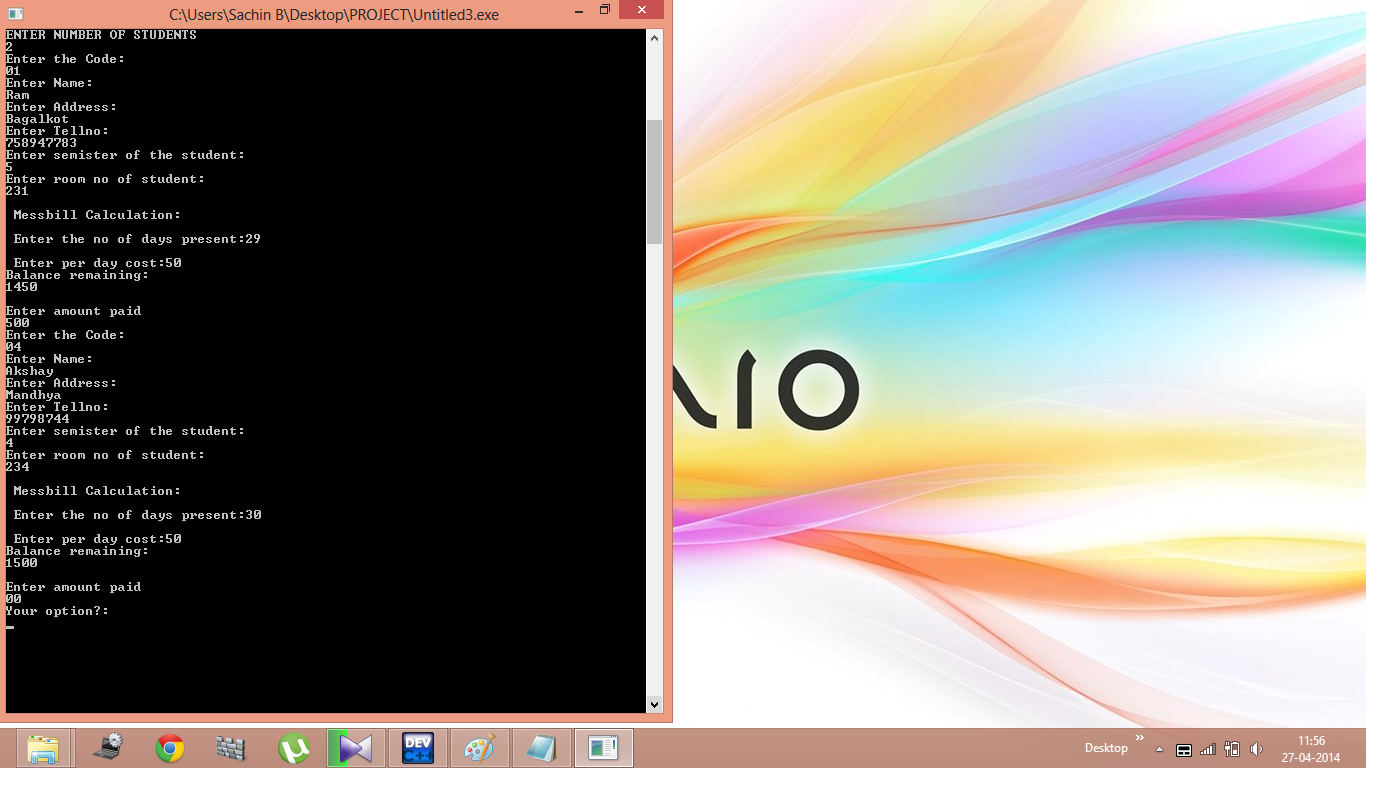
}

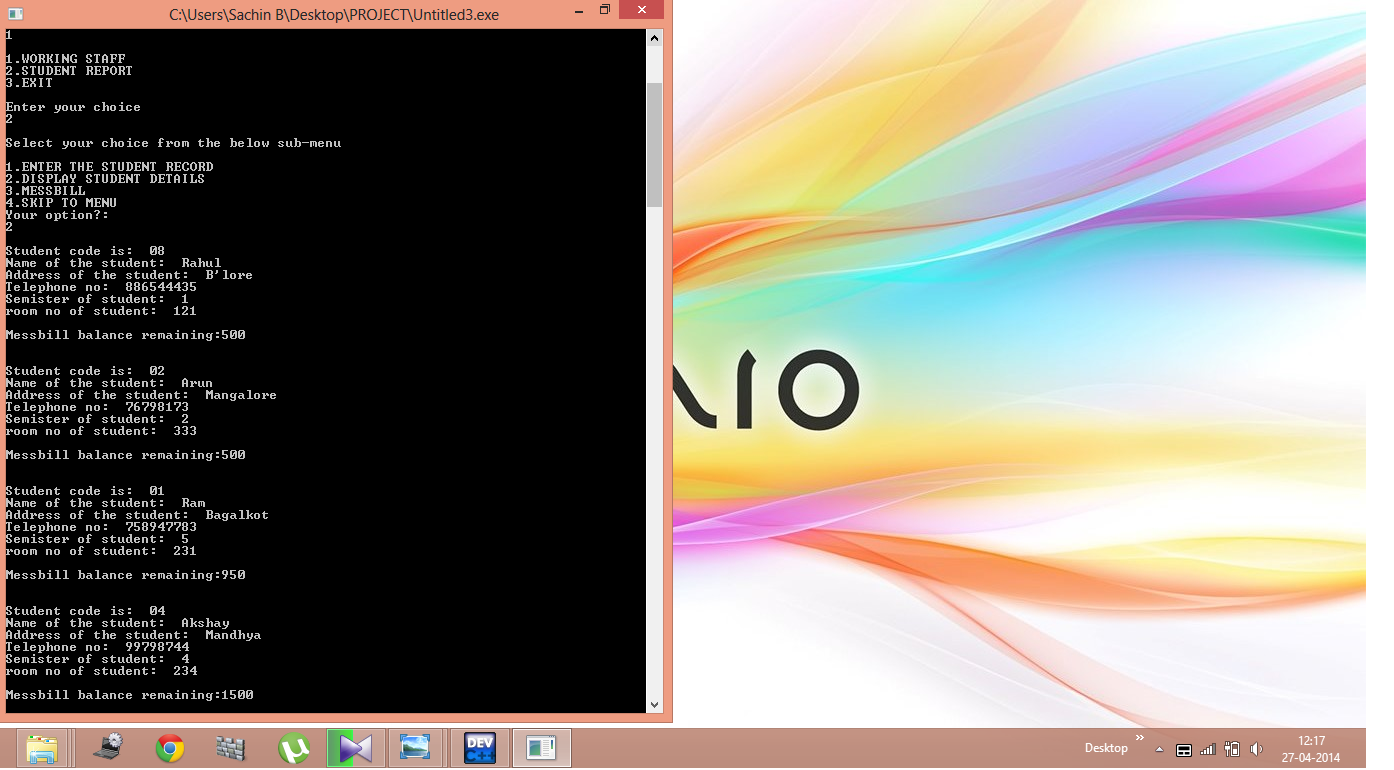
**SNAPSHOTS**

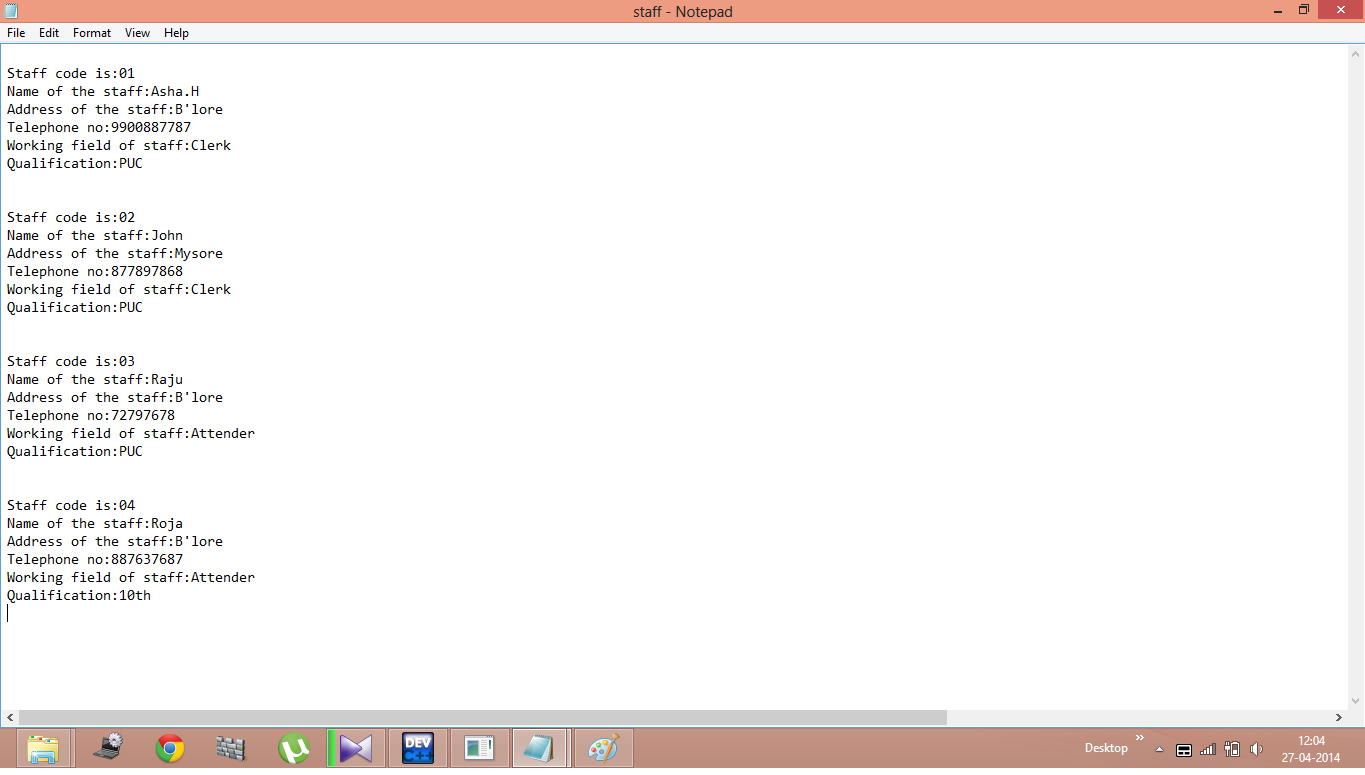


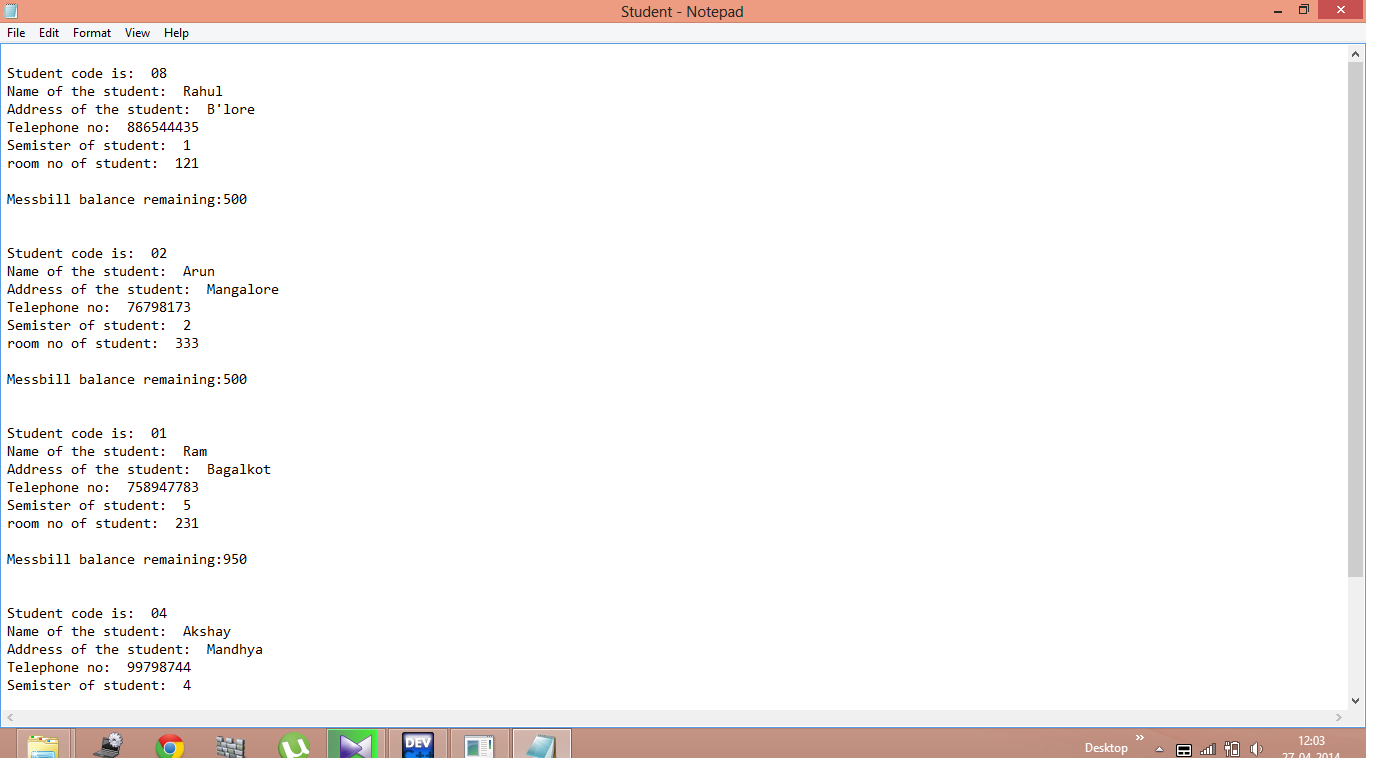












**CONCLUSION**

The above Hostel Management system software project is of great use for educational institutions especially for Hostels. It is also useful for all the students to make it as their academic mini and major project as student information management system.

**BIBLIOGRAPHY**

1. Complete references to C++ :By Herbert Schdilt.
2. Object Oriented Programming in C++ :By E.Balaguruswamy.
3. Programming with C++ :By R.Ravichandran.
4. The C++ Programming Language :By BjarneSttoustrup.
5. Effective C++ :By Scott Meyers.
6. Design Patterns :By Erinch Gamma, Richard Helm, Ralph Johnson.