

**HOTEL MANAGEMENT SYSTEM IN C++**

Hand OUt date: 17th august 2012

Hand IN date: 8th November 2012

**Submitted by:**

Prince Kumar

Intake no: PT1082219

# Certificate

This is to certify that the project Hotel Management System of the module Further Programming Concepts have been successfully completed by **Prince Kumar** Intake No.**PT1082219** of level 2.The project has been covered all the required areas in an appropriate manner.

**Acknowledgment**

*“Gratitude is the fairest blossom which springs from the soul”*

Feeling gratitude and not expressing it is like wrapping a present and not giving it. We take this opportunity to convey our heartfelt gratitude to each and every one who has supported us in every way or the other during the course of our project.

From the very core of our heart, we would like to express our sincere gratitude to Miss. Dipti for his supervisory guidance. We are always indebted to him for his kind support and constant encouragement and his enthusiasm to complete our project milestones.

It requires lots of efforts in terms of cooperation and support to fulfill various tasks involved during the project. We are always grateful to our peers and friends who have always encouraged us and guided us whenever we needed assistance.

We also take this opportunity to express our appreciation to all the participants involved during our preliminary research for their invaluable time to answer our queries and suggestions for the application to be developed.

Things always remain hidden in the shadow of the unsung heroes; still we would to thank all the people passively involved in the assignment, people who encouraged us day in and day out to make it a success.

At last but above all, we thank the Almighty for his blessings.

Prince Kumar

PT1082219

Table of Contents

[Certificate 2](#_Toc340143998)

[INTRODUCTION 6](#_Toc340143999)

[1.1 PURPOSE OF THE SYSTEM 6](#_Toc340144000)

[1.2 SCOPE 6](#_Toc340144001)

[1.3 HARDWARE AND SOFTWARE REQUIREMENTS 7](#_Toc340144002)

[2.0 Design Description 7](#_Toc340144003)

[2.1 Interface Design 7](#_Toc340144004)

[2.2 Class Diagram 14](#_Toc340144005)

[2.3 ERD 15](#_Toc340144006)

[2.4 DFD 16](#_Toc340144007)

[2.4.1 Context Diagram 16](#_Toc340144008)

[2.4.2 DFD-CHECK\_IN 17](#_Toc340144009)

[2.4.3. DFD-CHECK\_OUT 18](#_Toc340144010)

[3.0 Description & Justification of OO concepts 18](#_Toc340144011)

[3.1 Object 18](#_Toc340144012)

[3.2 Polymorphism 19](#_Toc340144013)

[3.3 Class 19](#_Toc340144014)

[3.3 Encapsulation 20](#_Toc340144015)

[4.0 Description of Validation 21](#_Toc340144016)

[5.0 Implementation 23](#_Toc340144017)

[6. TESTING 30](#_Toc340144018)

[7. CONCLUSION 32](#_Toc340144019)

[8. REFERENCES 32](#_Toc340144020)

# INTRODUCTION

A typical luxury Hotel requires a management system to control its various operations such as maintaining account of all the people in its domain of services, attending to various needs of customers and also achieving increased efficiency in the overall working of the Hotel itself.

## PURPOSE OF THE SYSTEM

The Hotel Management System aims to make simpler a staff’s interaction with the various modules of the Hotel and ease the process of acquiring information and providing services. The system can be accessed by the admin and customers but the highest priority given to admin that are allocated a login id and password. It will also allow cutting the operational costs of the hotel.

## SCOPE

In this system we will make extensive use of files system in C++.

We will have a **login id** system initially. In this system we will be having separate functions for

* Getting the information
* Getting customer information who are lodged in
* Allocating a room to the customer
* Checking the availability
* Displaying the features of the rooms.
* Preparing a billing function for the customer according to his room no.

In the software developed separate functions will be there for each of the above points so that there is ample scope for adding more features in the near future.

## 1.3 HARDWARE AND SOFTWARE REQUIREMENTS

The platform used is C++. Hence we decided to use Microsoft Visual Studio 2010/2008 edition (Professional Edition).

For optional coding and easy debugging we used Microsoft Visual Studio 2008 edition and made explicit .cpp files.

Henceforth for optimal usage of such software a windows based operating system preferably Windows vista/7/XP must be there. Also on the hardware part any system having Windows Vista/7/XP installed will suffice.

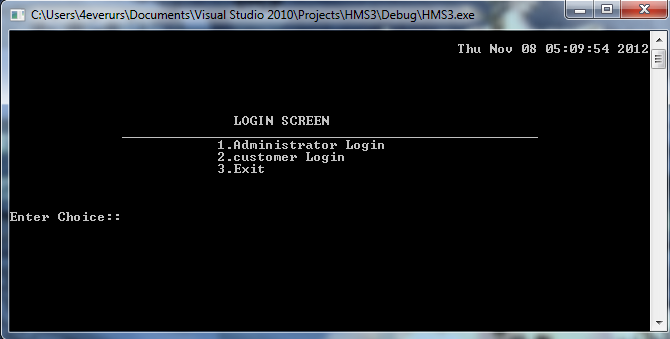
# Design Description

## 2.1 Interface Design

## 1.PNG

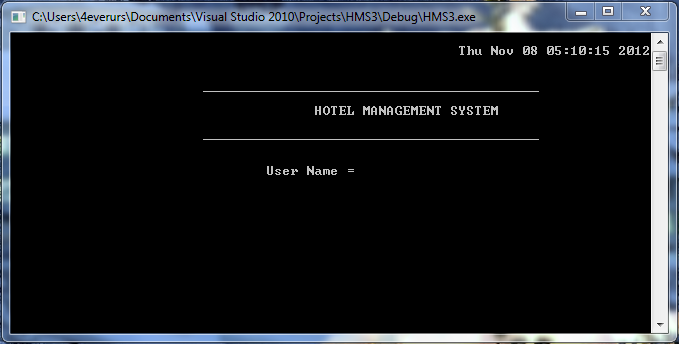
**Fig: welcome screen**

* **Welcome Page where user get the info about the application.**



**Fig : Login Screen**

* **Login screen for admin and for customer here admin have the highest priority where admin can add, delete, search, update and many more.**
* **Customer can view on availability of rooms**

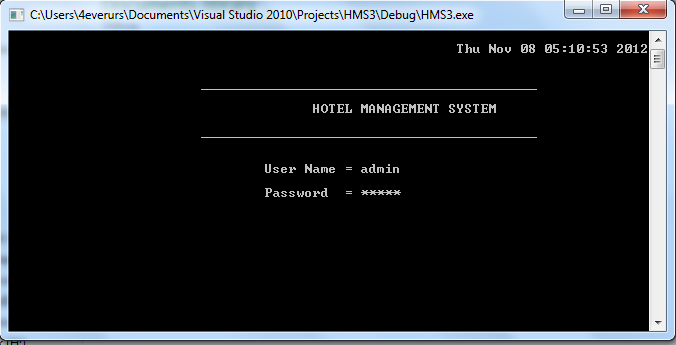
****

**Fig : Admin Login**

* **Admin :- User name : admin**

**Password: admin**

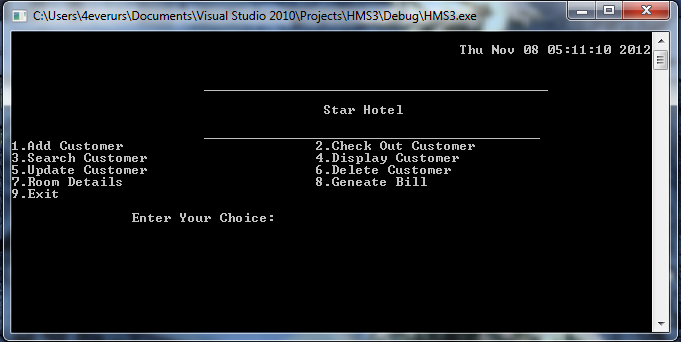
* **Login page for admin. it is for security purpose . no one can access admin page . only admin have the permission to handle records.**

****

**Fig : Admin Login**

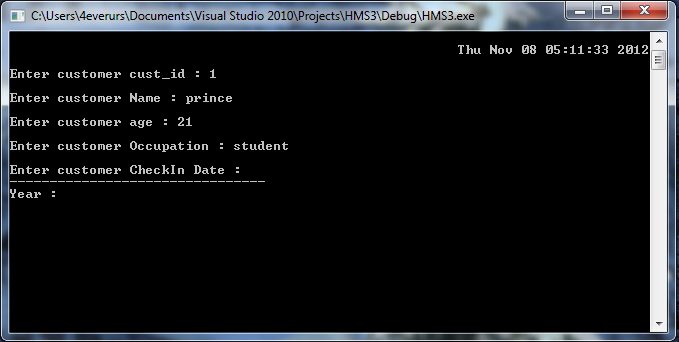
* **Admin :- User name : admin**

**Password: admin**

****

**Fig : Admin screen**

* **Options to handle records.**

****

**Fig : Add\_customer**

* **Adding customer details**

**Name**

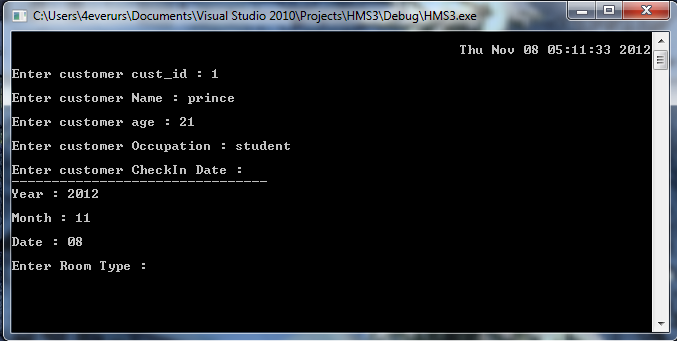
**ID**

**ID\_Proof**

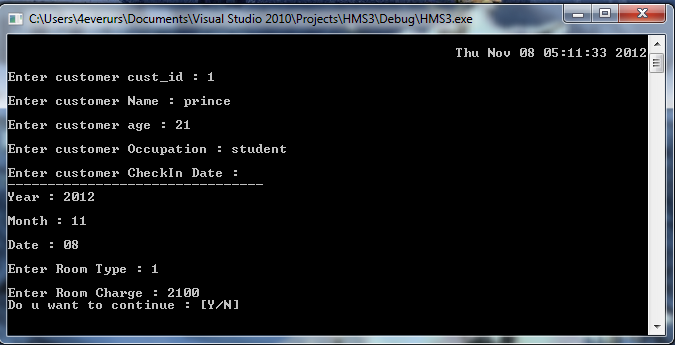
**Customer\_occupation**

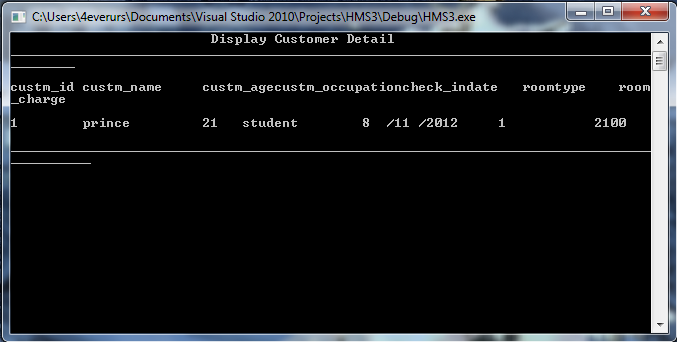
**Customer\_checkin**

**Customer\_checkout**

****

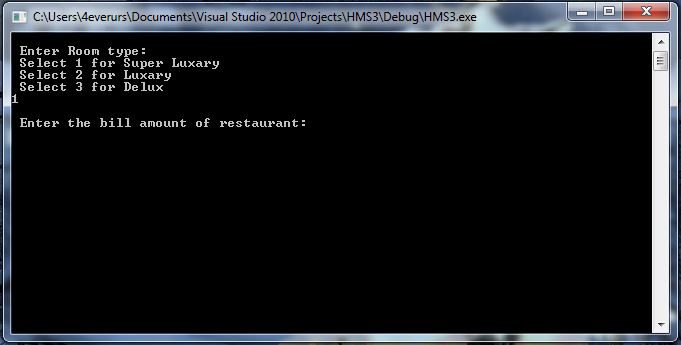
**Fig : Add\_customer1**

****

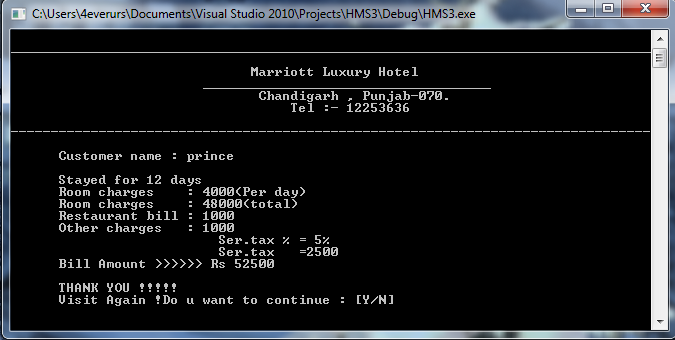
****

**Fig : Display\_Customer**

* **Displaying the details of the customer**

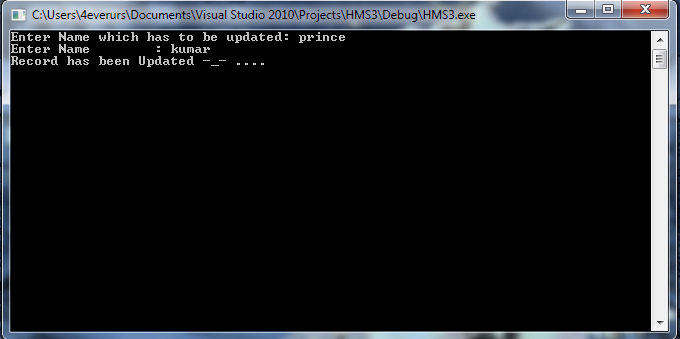
****

**Fig : Customer\_bill**

****

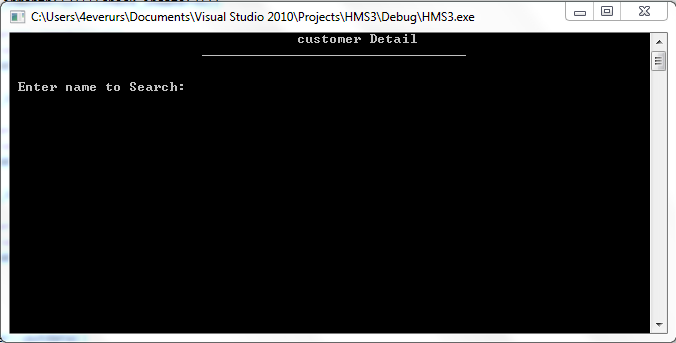
**Fig : Final\_Bill**

* **Calculating the overall bill by taking Cust\_name , no. of stayed , room charges , other charges**

****

**Fig : Update\_customer**

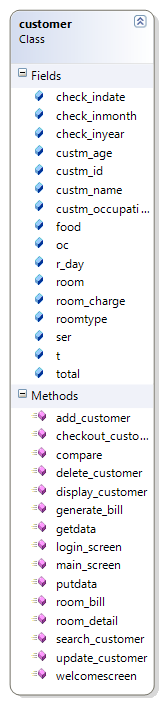
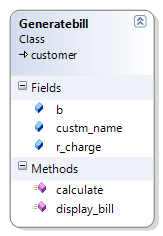
* **Updating the details of the customer by taking user name.**

****

**Fig : search\_customer**

* **Customer can be searched by just taking the name of the customer**

## 2.2 Class Diagram

Public

# 

# 2.3 ERD

Password

User name

**ADMIN**

(1,1)

Check

( 1,n )

Custm\_checkout

**Customer**

Custm\_checkin

Custm\_Id

Custm\_occupation

Custm\_name

Custm\_Idproof

Custm\_age

## 2.4 DFD

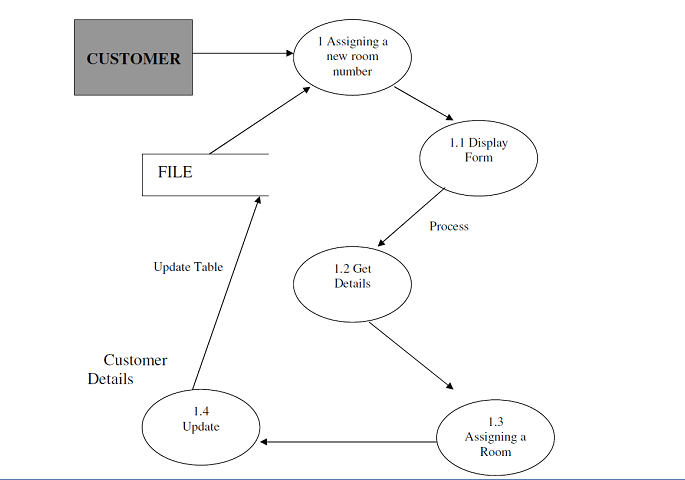
### 2.4.1 Context Diagram

CUSTOMER CHECK\_OUT

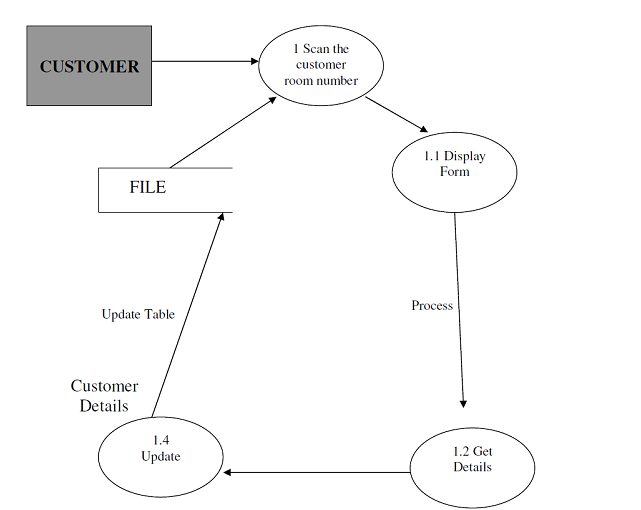
CUSTOMER

ROOM

### 2.4.2 DFD-CHECK\_IN



### 2.4.3. DFD-CHECK\_OUT



# Description & Justification of OO concepts

## Object

We used to create object to call the function which is related to any particular class we had objects to call the function in this assignment. here we had created view object of class customer to call the function welcomecreen() and loginscreen() functions to call then in main functions

#include"customer.h"

using namespace std;

void main()

{

customer view;

view.welcomescreen();

view.login\_screen();

}

## Polymorphism

Complier is able to select the appropriate function for a particular call at the complie time itself (also called as early binding )

we use polymorphism to access the variables from different class in this project we had use polymorphism to get access the variables of customer class.

**Code:**

class Generatebill:public customer

## 3.3 Class

Bind the data and its associated functions together it allows data to be hidden if necessary from external use defining a class we are creating a new abstract data type that can be treated like other built – in data type

Code :

#ifndef H\_M

#define H\_M

#include<string>

#include<iomanip>

#include<iostream>

#include<fstream>

#include<conio.h>

#include<math.h>

#include<time.h>

using namespace std;

class **customer**

{

public:

int custm\_id;

int custm\_age;

int check\_inyear;

int check\_inmonth;

int check\_indate;

char roomtype[20];

char custm\_name[30];

int room\_charge;

int compare(char[]);

char custm\_occupation[20];

void getdata();

void putdata();

long total; //variable for Generate Bill

int food;

long room;

float ser;

int t;

int oc;

int r\_day;

public:

void welcomescreen();

void login\_screen();

void add\_customer();

void delete\_customer();

void update\_customer();

void search\_customer();

void main\_screen();

void display\_customer();

void checkout\_customer();

void room\_detail();

void generate\_bill();

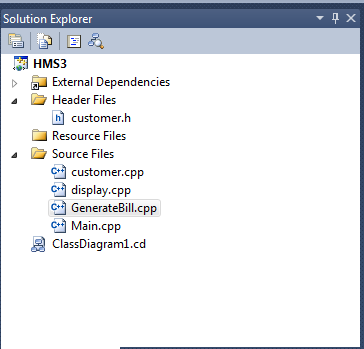
void room\_bill();

};

#endif

## Encapsulation

It is a process of wrapping up to data value and member in a single unit (class).



# 4.0 Description of Validation

In every project validation is the important parts while the development. In Hotel Management System we had also provided proper validation to make system error free. In this system we had provided validation in date month and year to get proper date from administrator. The codes which we had used to get proper validation in date system are mentioned below.

**code:**

cout<<"\nEnter customer CheckIn Date : ";

START:

cout<<"\n--------------------------------";

cout<<"\nYear : ";

cin>>check\_inyear;

cout<<"\nMonth : ";

cin>>check\_inmonth;

cout<<"\nDate : ";

cin>>check\_indate;

cout<<"\nEnter Room Type : ";

cin>>roomtype;

cout<<"\nEnter Room Charge : ";

cin>>room\_charge;

if((check\_inmonth==2) && (check\_indate>29))

{

cout<<"wrong input!!!\n";

cout<<"\nenter the date again....\n";

goto START;

}

if((check\_inmonth>12)||(check\_indate>31))

{

cout<<"wrong input!!!\n";

cout<<"\nenter the date again....\n";

goto START;

}

if((check\_inmonth==4||check\_inmonth==6||check\_inmonth==9||check\_inmonth==11) && (check\_indate>30))

{

cout<<"wrong input!!!\n";

cout<<"\nenter the date again....\n";

goto START;

}

if((check\_inyear%4)!=0 && (check\_inmonth==2) && (check\_indate>28))

{

cout<<"wrong input!!!\n";

cout<<"\nenter the date again....\n";

goto START;

}

We had use the validation in the adminstration login while entering the wrong user name & password it show the popup menu.

cout<<"\t\t\t\tUser Name = ";

cin>>username;

if(strcmp(username,"admin")==0)

{

string pass ="";

char ch;

cout <<"\n\t\t\t\tPassword = ";

ch = \_getch();

while(ch != 13)

{

pass.push\_back(ch);

cout << '\*';

ch = \_getch();

}

if(pass == "admin")

{

cout <<"\n\t\t"<< "Access granted";

do

{

**}**

**}**

# Implementation

**WELCOME SCREEN:**

void customer::welcomescreen()

{

system("cls");

cout<< endl << endl << endl<< endl;

cout<<" COMPUTERISED MANAGEMENT SYSTEM "<<endl;

cout<<" APIIT SD INDIA, PANIPAT "<< endl;

cout<<" INDIVIDUAL ASSINGMENT "<< endl;

cout<<" ON "<< endl;

cout<<" Further Programming Concepts in C++ "<< endl;

cout<<" (CE00314-2) "<<endl;

cout<<" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<< endl;

cout<< endl << endl << endl<< endl;

cout<<" SUBMITTED BY: SUBMITTED TO: "<<endl;

cout<<" Prince kumar Ms. Deepti Dhingra "<<endl;

cout<<" (PT082219) (LECTURER) "<<endl;

cout<< endl << endl;

cout<<" Press Enter to continue...." << endl << endl << endl << endl;

getch();

}

**LOGIN SCREEN.**

void customer::login\_screen()

{

time\_t t;

system("cls");

t=time(NULL);

cout<<"\n "<<ctime(&t);

customer c;

t -= 24L\*60L\*60L;

char choice;

cout<< endl << endl << endl<< endl;

cout<<" LOGIN SCREEN\n";

cout<<" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<< endl;

cout<<" 1.Administrator Login\n";

cout<<" 2.customer Login\n";

cout<<" 3.Exit\n";

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

cout<<"Enter Choice:: ";

cin>>choice;

switch(choice)

{

case'1':

system("cls");

c.main\_screen();

break;

case '2':

search\_customer();

break;

case'3':

exit(0);

break;

default:

cout<<"wrong choice entered";

}

}

**ADD CUSTOMER**

void customer::add\_customer()

{

customer c;

ofstream outf;

outf.open("customer.txt",ios::out|ios::app|ios::binary);

c.getdata();

outf.write((char\*)&c,sizeof(c));

outf.close();

\_getch();

}

**SEARCH CUSTOMER**

void customer::search\_customer()

{

fstream file;

customer c;

int i=1;

char nam[25];

file.open("customer.txt",ios::in);

cout<<"\t\t\t\t customer Detail \n";

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

cout<<"\t\t\t\t\n Enter name to Search: ";

cin>>nam;

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

while(file.read((char \*) &c, sizeof(customer)))

{

if(c.compare(nam))

{

cout<<"\t\t\t\t custm\_id"<<" "<<"custm\_name \n"<<endl;

cout<<" "<<c.custm\_id<<" "<<c.custm\_name ;

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

cout<<"\n\t\t\t ...Thank you for using ^\_^ ... ";

i=0;

break ;

}

}

file.close() ;

if(i)

{

cout<<"Sorry!!!! Record Not found";

}

getch();

}

**DELETE CUSTOMER**

void customer::delete\_customer()

{

fstream file;

fstream temp;

temp.open("tempdata.txt",ios::out);

customer c;

int i=1;

char nam[25];

file.open("customer.txt",ios::in);

cout<<"\t\t\t\t All customer Detail \n";

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

cout<<"\t\t\t\t\n Enter name to Search: ";

cin>>nam;

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

while(file.read((char \*) &c, sizeof(customer)))

{

if(!c.compare(nam))

{

cout<<"\t\t\t\t custm\_id"<<" "<<"custm\_name\n"<<endl;

cout<<" "<<c.custm\_id<<" "<<c.custm\_name;

i=0;

temp.write((char \*)&c,sizeof(customer));

}

}

file.close() ;

temp.close();

remove("customer.txt");

rename("tempfile.txt","customer.txt");

getch();

}

**CHECKOUT\_CUSTOMER**

void customer::checkout\_customer()

{

system("cls");

cout<<"\n\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"\n\t\t\t\t HOTEL MANAGEMENT SYSTEM "<<endl;

cout<<"\n\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n"<<endl;

cout<<"\t\t\t\t All customer Detail \n";

}

**ROOM\_DETAIL**

{

system("cls");

cout<<"\n\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"\n\t\t\t\tHOTEL MANAGEMENT SYSTEM "<<endl;

cout<<"\n\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"<<endl;

cout<<"\t\t\t\tAll customer Detail \n";

cout<<left<<setw(10)<<"roomtype"<<left<<setw(15)<<"room\_charge"<<endl;

}

**Generate Bill**

#include "customer.h"

class Generatebill:public customer

{

public :

int r\_charge;

char custm\_name[25];

char b;

public:

void calculate()

{

system("cls");

cout<<"\n Enter Room type:";

cout<<"\n Select 1 for Super Luxary \n Select 2 for Luxary \n Select 3 for Delux \n";

cin>>b;

if(b=1)

{

r\_charge=4000;

}

else

if(b=2)

{

r\_charge=3000;

}

else

if(b=3)

{

r\_charge=2000;

}

else{

cout<<"\n Wrong choice ...";

cout<<"\n ERROR: Room is not allocated.";

cout<<"\n Bill cannot be created ";

}

cout<<"\n Enter the bill amount of restaurant:";

cin>>food;

cout<<"\n Enter other charges:";

cin>>oc;

cout<<"\n Enter customer name:";

cin>>custm\_name;

cout<<"\n Enter number of days:";

cin>>r\_day;

room= r\_charge \* r\_day;

t=(int) room + food + oc;

ser=(int)(0.05\*t);

total=t+ser;

}

void display\_bill()

{

system("CLS");

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

cout<<"\n Marriott Luxury Hotel ";

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

cout<<"\n Chandigarh , Punjab-070. ";

cout<<"\n Tel :- 12253636 ";

cout<<"\n--------------------------------------------------------------------------------";

cout<<"\n Customer name : "<<custm\_name;

cout<<"\n ";

cout<<"\n Stayed for "<<r\_day<<" days";

cout<<"\n Room charges : "<<r\_charge<<"(Per day)";

cout<<"\n Room charges : "<<room<<"(total)";

cout<<"\n Restaurant bill : "<<food;

cout<<"\n Other charges : "<<oc;

cout<<"\n Ser.tax % = 5%";

cout<<"\n Ser.tax ="<<ser;

cout<<"\n Bill Amount >>>>>> Rs "<<total;

cout<<"\n ";

cout<<"\n THANK YOU !!!!!";

cout<<"\n Visit Again !";

}

};

void customer::generate\_bill()

{

Generatebill b;

b.calculate();

b.display\_bill();

}

# 6. TESTING

TEST CASE 1:

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE Number | CASE 1.0 | TEST CASE | LOGIN |
| SYSTEM | HMS | SUBSYSTEM | NIL |
| DESIGNED BY | PRINCE | DESIGN DATE | 1 NOV |
| EXECUTED BY | PRINCE | EXECUTION DATE | 8 NOV |
| SHORT DESCRIPTION | Administration should login the system and display the transaction  Screen on input of all choice as the username and 'a' as password | | |

|  |
| --- |
| PRE - CONDITIONS :  The user of the system is registered in the HMS and has a valid username and password.  User inputs 'admin' as the username and 'admin' as the password.  System displays the transaction screen |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STEPS | ACTION | EXPECTED RESPONSE | PASS / FAIL | COMMENT |
| 1 | Go to login page | Login page is popped in the centre of the window | PASS | Login page is successfully displayed |
| 2 | Enter required information | Input 'admin' as the username and 'admin' as the password | PASS |  |
| 3 | Submit the information | The user presses the enter button and waits for the reply | PASS | System processes the data |
| 4 | Display main page | The application displays the transaction screen granting successful login to the user | PASS | Transaction screen is Displayed |

TEST CASE 2:

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE Number | CASE 2.0 | TEST CASE | Issue a Book |
| SYSTEM | LMS | SUBSYSTEM | NIL |
| DESIGNED BY | PRINCE | DESIGN DATE | 1 OCTOBER |
| EXECUTED BY | PRINCE | EXECUTION DATE | 9 OCTOBER |
| SHORT DESCRIPTION | HSM should ADD CUSTOMER on successful entry of required information | | |

|  |
| --- |
| PRE - CONDITIONS :  HMS must ADD CUSTOMER on providing necessary information such as cutomer id, customer name, occupation etc. and must successfully add the database accordingly. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STEPS | ACTION | EXPECTED RESPONSE | PASS / FAIL | COMMENT |
| 1 | Select add customer no | Add customer page should be displayed | PASS | Add customer  page is successfully displayed |
| 2 | Enter required information | Provide customer id, name and of the customer you want to add | PASS | No error found |
| 3 | Submit the information | The user enter and waits for the messege | PASS | System processes the data |
| 4 | Add customer | The application should add customer and updated the database accordingly | FAIL | Customer were not updated in the database |

TEST CASE 3:

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE Number | CASE 3.0 | TEST CASE | Return a Book |
| SYSTEM | HMS3 | SUBSYSTEM | NIL |
| DESIGNED BY | PRINCE | DESIGN DATE | 1 OCTOBER |
| EXECUTED BY | PRINCE | EXECUTION DATE | 10 OCTOBER |
| SHORT DESCRIPTION | HMSshould calculate the room charges entry of required information | | |

|  |
| --- |
| PRE - CONDITIONS :  HMSmust calculate on providing necessary information such as room name, room fare, extra charge etc. and must successfully show the total charge accordingly. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STEPS | ACTION | EXPECTED RESPONSE | PASS / FAIL | COMMENT |
| 1 | Go to screen page | calculate should be displayed | PASS | Calculate page successfully displayed |
| 2 | Enter required information | Room type, room fare ,extra charges of the customer you want to charge | PASS | No error found |
| 3 | Show the information | The user enter and waits for the display | FAIL | System not properly calculate |

# CONCLUSION

At the end of this coursework, I was able to:

* Explain object oriented programming concepts and apply them to the modelling of real world systems.
* Explain the object oriented paradigm and utilisation of the offered facilities.
* Demonstrate the ability to develop and derive new class structures and organise them such that they will model real world systems within computers.

# REFERENCES

* 1. Balaguruswamy 1 E,(2006),” Object Oriented Programming with C++”. 3rd ed. Tata McGraw-Hill Publishing Company Limited Pages 54 - 68.
  2. http://v2.cplusplus.com/forum/lounge/12685/ (CODER)