|  |
| --- |
| http://cdn3.howtogeek.com/wp-content/uploads/gg/up/sshot4e491a8aa5e55.jpg |
| **Computer Project Report**  **Hotel Management** |
| |  |  |  | | --- | --- | --- | |  |  |  | |
|  |

**Table of Contents**

* [**Certificate** 2](#_Toc435546988)
* [**Acknowledgements** 3](#_Toc435546989)
* [**Synopsis** 4](#_Toc435546990)
* [**Main contents of the Project** 5](#_Toc435546991)
* [**Source Code** 6](#_Toc435546992)
* [**Output** 25](#_Toc435546993)

# **Synopsis**

The following program is based on a ‘**Hotel Management system’**. The program first accepts a login password for an employee to log in.  
When the login is successful, the main menu is displayed which contains options for customer details and food details.   
In case of wrong password, the menu is not displayed.

Customer details include functions like adding a customer to database, noting his arrival and the type of room the customer wishes to stay in. Arrangement for modifying the above mentioned data has been provided along with display function which shows the current list of people staying in the hotel.

Food details include the food menu, from which the customer can satisfy his appetite. Provision for deleting particular unavailable items or clearing the menu has been made.

Room bill and food bill have been calculated accordingly (acc. to customer’s duration of stay, food that the customer has ordered)

Consequently, the customer’s total bill has been calculated.

# **Main contents of the Project**

**Class 1 : Food()**

**Functions** : 1)food\_app() (Add food items)

2) food\_menu() (Display main menu)

3) food\_del() (Delete items from menu)

4) food\_disp() (Display food menu)

5) food\_adel() (Delete entire menu)

**Class 2 : Customer() (PUBLICLY derived from class food)**

**Functions** : 1) cust\_bill() (Calculate customer bill)

2) room\_bill() (Calculate room bill)

3) food\_bill() (Calculate food bill)

4) total\_bill() (Calculate total bill)

5) cust\_detail() (Main menu for customer details)

6) cust\_app() (Add a customer to database)

7) cust\_mod() (Modify customer details)

8) cust\_del() (Delete a customer from database)

9) cust\_adel() (Delete entire customer database)

10) cust\_disp() (Display customer details)

**Class 4 : Control() (Contains objects of other classes)**

**Functions** : 1) pass() (For password approval)

2) mmenu() (Show main menu)

3 )information (Standard message)

# **Source Code**

#include<iostream>

#include<string.h>

#include<stdio.h>

#include<fstream>

#include<iomanip>

#include<stdlib.h>

#include<conio.h>

using namespace std;

fstream f; //Global fstream object

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Class-1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class containing food functions

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class food

{

public:

food()

{

foodobj.sno = 0;

}

void food\_app(); //Function prototype 1(Fnc 1)

void food\_menu(); //Function prototype 2(Fnc 2)

void food\_del(); //Function prototype 3(Fnc 3)

void food\_disp(); //Function prototype 4(Fnc 4)

void food\_adel(); //Function prototype 5(Fnc 5)

protected:

struct fd

{

int sno;

char name[55];

float price;

}foodobj;

}; //End of class food

void food::food\_app() //Function to append in food menu

{

system("cls");

int tell;

f.open("food.txt", ios::app | ios::binary); //Existing file opened

f.seekp(0, ios::end); //File pointer placed at end to add data

tell = f.tellp() / sizeof(foodobj); //Obtaining no. of records

foodobj.sno = tell+1;

cin.get();

cout << "\n\tAdding food items.\n"

<< "\n Enter the item name.\t";

gets\_s(foodobj.name, 54);

cout << " Enter the price.\t";

cin >> foodobj.price;

f.write((char\*)&foodobj, sizeof(foodobj)); //Writing data on file

f.close(); //Closing link with food.txt

cout << "\n Data successfully entered...\n ";

system("pause");

}

void food::food\_menu() //Function for displaying food menu

{

int choice = 0;

do

{

system("cls");

cout <<"\n\tFood Menu\n"

<< "\n 1. Add food items."

<< "\n 2. Delete a food item."

<< "\n 3. Delete all food items."

<< "\n 4. View available food items."

<< "\n 5. Go to Main Menu.\n"

<< "\n Enter choice : ";

cin >> choice;

switch (choice)

{

case 1: food\_app();

break;

case 2: food\_del();

break;

case 3: food\_adel();

break;

case 4: food\_disp();

break;

case 5: break;

default: cout << "\n Wrong choice !!! Enter correct option.\n ";

system("pause");

}

} while (choice != 5);

}

void food::food\_del() //Function to delete a particular record in food menu

{

ifstream fin;

ofstream fout;

int no, ctr = 0;

cout << "\n Enter item number to be deleted :\t";

cin >> no;

fin.open("food.txt", ios::binary); //File opened for reading

fout.open("temp.txt", ios::binary); //File for writing temporary data

while (fin.read((char \*)&foodobj, sizeof(foodobj)))

{

if (no == foodobj.sno) //Checking for match

{

ctr = 1;

cout << "\n Record successfully deleted.\n ";

//If customer no. matches, it is not written on temp

}

else

fout.write((char \*)&foodobj, sizeof(foodobj));

}

fin.close();

fout.close();

remove("food.txt");

rename("temp.txt", "food.txt");

if (ctr == 0)

cout << "\n Item number not found!!!\n ";

system("pause");

}

void food::food\_adel() //Function to empty food menu

{

int choice;

do

{

system("cls");

cout << "\n Are you sure you want to delete all data???\n"

<< "\n 1. Yes"

<< "\n 2. No\n"

<< "\n Enter choice : ";

cin >> choice; //Confirming to empty food menu

switch (choice)

{

case 1: remove("food.txt"); //File deleted

f.open("food.txt", ios::out);

//File created again to show that it is empty

foodobj.sno = 0;

f.close();

cout << "\n Data successfully deleted.\n ";

system("pause");

break;

case 2: cout << " Data not deleted.\n ";

system("pause");

break;

default: cout << "\n Wrong choice!!! Enter a valid option...\n";

break;

}

} while (choice != 1 && choice != 2);

} //End of functions of class food

void food::food\_disp() //function to display food items

{

system("cls");

cout << "\n\t\t Available Items...\n";

f.open("food.txt", ios::in | ios::binary);

f.seekg(0, ios::beg); //Bring file pointer to beginning

if (foodobj.sno == 0)

cout << "\n No records found!!!\n ";

else

{

cout << "\n Item No. Food name Price\n";

while (f.read((char\*)&foodobj, sizeof(foodobj))) //Read from beginning

{

cout << "\n " << foodobj.sno

<< " " << foodobj.name

<< " " << foodobj.price;

}

}

f.close(); //Closing link with food.txt

cout << "\n ";

system("pause");

}

/\*\*\*\*\*\*\*\*\*\*\*\* Class 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class containing customer data

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class customer: public food //Class customer inherits PUBLICLY from class food

{

private:

int q, w;

struct cust

{

int c\_no,

a\_date,

a\_month,

a\_year,

d\_date,

d\_month,

d\_year,

room\_no;

char c\_name[21],

room\_type[25];

long double room\_bill,

food\_bill;

}p; //p is object of structure cust

public:

customer() //constructor to intialize private members

{

p.c\_no = 0;

p.d\_date = 0;

p.d\_month = 0;

p.room\_bill = 0;

p.food\_bill = 0;

}

void cust\_bill();

void room\_bill();

void food\_bill();

void total\_bill();

void cust\_detail();

void cust\_app();

void cust\_mod();

void cust\_del();

void cust\_adel();

void cust\_disp();

}; //End of class customer

void customer::cust\_bill() //Function for calculation customer bill

{

int choice;

do

{

system("cls");

cout << "\n Customer Billing\n"

<< "\n 1. Calculate room bill."

<< "\n 2. Calculate food bill."

<< "\n 3. View customer bill."

<< "\n 4. Go to Main Menu\n"

<< "\n Enter choice : ";

cin >> choice;

switch (choice)

{

case 1: room\_bill();

break;

case 2: food\_bill();

break;

case 3: total\_bill();

break;

case 4: break;

default: cout << "\n Wrong choice !!! Enter correct option.\n ";

system("pause");

break;

}

} while (choice != 4);

}

void customer::room\_bill() //Function for room bill

{

int days, cno, dt, mth, yr, valid = 0, found = 0, yrdiff, mthdiff;

long pos;

long double bill;

system("cls");

cout << "\n\t\tCalculating room bill...\n"

<< "\n Enter customer no. of departing customer:\t";

cin >> cno;

cout << "\n";

f.open("cust.txt", ios::binary | ios::in | ios::out);

while (!f.eof())

{

pos = f.tellg();

f.read((char\*)&p, sizeof(p));

if (cno == p.c\_no) //Checking for customer ID match

{

found = 1;

do //accepted departure year

{

valid = 0;

cout << " Year of departure(after 1980) : ";

cin >> yr;

if (yr < 1980 || yr > 2016) //Checking incorrect entry

{

cout << "\n Wrong entry.... Enter a valid year.\n ";

system("pause");

cout << "\n";

++valid;

}

} while (valid != 0);

do //Departure Month accepted

{

valid = 0;

cout << " Month of departure : ";

cin >> mth;

if (mth < 1 || mth > 12) //Checking incorrect entry

{

cout << "\n Wrong entry.... Enter a valid month.\n ";

system("pause");

cout << "\n";

++valid;

}

} while (valid != 0);

do //Departure date accepted

{

valid = 0;

cout << " Date of departure : ";

cin >> dt;

if (dt < 1 || mth == 2 && (((yr % 4 == 0) && dt>29) || ((yr % 4 != 0) && dt > 28)) || dt > 31)

{

cout << "\n Wrong entry.... Enter a valid date.\n ";

system("pause");

cout << "\n";

++valid;

}

} while (valid != 0);

yrdiff = yr - p.a\_year;

mthdiff = (yrdiff \* 12) + (mth - p.a\_month);

days = (mthdiff \* 30) + (dt - p.a\_date); //calc. no of days of stay

if (\_strcmpi(p.room\_type, "Deluxe") == 0) //calc. bill acc to room

type

{

bill = days \* 500;

}

else if (\_strcmpi(p.room\_type, "Suite ") == 0)

{

bill = days \* 1000;

}

else if (\_strcmpi(p.room\_type, "Luxury") == 0)

{

bill = days \* 1500;

}

p.d\_date = dt;

p.d\_month = mth;

p.d\_year = yr;

p.room\_bill = bill;

cout << "\n Customer No. Name Room Type Arrival

Departure Days Bill\n"

<< "\n " << p.c\_no << " "

<< p.c\_name << "\t"

<< p.room\_type << " "

<< p.a\_date << " / " << p.a\_month << " / " << p.a\_year << " ";

if (p.d\_date == 0)

cout << " No data available";

else

cout << " " << p.d\_date << " / " << p.d\_month << " / " <<

p.d\_year;

cout << " " << days

<< " " << p.room\_bill << " \n\n ";

f.seekg(pos);

f.write((char \*)&p, sizeof(p));

break;

} //end of if

} //end of while

if (found == 0)

{

cout << "\nCustomer number not found!!!\n";

}

f.close();

system("pause");

} //End of room bill

void customer::food\_bill() //Function for calculation of Food bill

{

fstream f1, f2;

long pos;

int cno, qty, ctrc = 0, ctrf = 0;

char itemname[55];

system("cls");

f1.open("cust.txt", ios::binary | ios::in | ios::out);

f2.open("food.txt", ios::binary | ios::in | ios::out);

cout << "\n\t\tCalculating food bill...\n"

<< " Enter customer number.\t";

cin >> cno;

while (!f1.eof())

{

pos = f1.tellg();

f1.read((char\*)&p, sizeof(p));

if (cno == p.c\_no) // Checking for match

{

cin.get();

cout << " Enter name of the food item purchased.\t";

gets\_s(itemname, 54);

while (f2.read((char\*)&foodobj, sizeof(foodobj)))

{

if (\_strcmpi(foodobj.name, itemname) == 0)

{

cout << " Enter the quantity.\t";

cin >> qty;

p.food\_bill = qty\*foodobj.price; //Calculating bill

cout << "\n Customer No. Name Food Name

Quantity Bill\n";

cout << "\n " << p.c\_no << " "

<< p.c\_name << "\t "

<< foodobj.name << "\t\t "

<< qty << " "

<< p.food\_bill;

f1.seekg(pos);

f1.write((char \*)&p, sizeof(p)); //Writing modified data

ctrf = 1;

break;

}

}

if (ctrf == 0)

cout << "\n You entered a wrong item name!!!\n";

ctrc = 1;

break;

}

}

if (ctrc == 0)

cout << "\n You entered a wrong customer number!!!\n";

f1.close();

f2.close();

cout << "\n\n ";

system("pause");

}

void customer::total\_bill() //Function for calculation of Total bill

{

system("cls");

long double total\_bill = 0;

cout << " Customer No. Name Room Bill Food Bill Total

Bill\n";

f.open("cust.txt", ios::binary | ios::in | ios::out);

while (f.read((char\*)&p, sizeof(p)))

{

total\_bill = p.room\_bill + p.food\_bill; //Calculating total bill here

cout << "\n " << p.c\_no << " "

<< p.c\_name << "\t";

if (p.room\_bill == 0)

cout << "No data available ";

else

cout <<" "<< p.room\_bill << " ";

if (p.food\_bill == 0)

cout << p.food\_bill << " ";

else

cout << p.food\_bill << " ";

cout << total\_bill;

}

f.close();

cout << "\n\n Thank You for visiting us... It was an honor serving you.\n Please do

visit again.\n\n ";

system("pause");

}

void customer::cust\_detail() //Function for displaying menu for customer

{

int c;

do

{

system("cls");

cout << "\n Customer Menu\n"

<< "\n 1. Add new record."

<< "\n 2. Modify a record."

<< "\n 3. Delete a record. "

<< "\n 4. Delete all records."

<< "\n 5. Display all records."

<< "\n 6. Go to Main menu\n"

<< "\n Enter choice : ";

cin >> c;

switch (c)

{

case 1: cust\_app();

break;

case 2: cust\_mod();

break;

case 3: cust\_del();

break;

case 4: cust\_adel();

break;

case 5: cust\_disp();

break;

case 6: break;

default: cout << "\n Wrong choice!!! Enter a correct option.\n ";

system("pause");

}

} while (c != 6);

} //End of customer detail functions

void customer::cust\_app() //Function to append customer details

{

system("cls");

fstream f1;

int tell, valid, tempdt, tempmon, tempyr, choice;

char tempnm[21];

f1.open("cust.txt", ios::app | ios::binary);

//File opened in append mode to add data

f1.seekp(0, ios::end); //File pointer placed at end

tell = f1.tellp() / sizeof(p); //Obtaining total no. of records present

cin.get();

cout << "\n\tAdding customer data...\n"

<< " Name : ";

gets\_s(tempnm, 20);

do

{

valid = 0;

cout << "\n Year of arrival (after 1980) : ";

cin >> tempyr;

if (tempyr < 1980 || tempyr > 2016) //Checking incorrect entry

{

cout << "\n Wrong entry!!! Enter a valid year.\n ";

system("pause");

++valid;

}

} while (valid != 0);

do

{

valid = 0;

cout << "\n Month of arrival : ";

cin >> tempmon;

if (tempmon < 1 || tempmon > 12) //Checking incorrect entry

{

cout << "\n Wrong entry!!! Enter a valid month.\n ";

system("pause");

++valid;

}

} while (valid != 0);

do

{

valid = 0;

cout << "\n Date of arrival : ";

cin >> tempdt;

if (tempdt < 1 || tempmon == 2 && (((tempyr % 4 == 0) && tempdt>29) ||

((tempyr % 4 != 0) && tempdt > 28)) || tempdt > 31)

{

cout << "\n Wrong entry!!! Enter a valid date.\n ";

system("pause");

++valid;

}

} while (valid != 0);

do //This accepts room type acc to customer

{

cout << "\n ROOM TYPES : "

<< "\n 1. Deluxe : Rs. 500"

<< "\n 2. Suite : Rs. 1000"

<< "\n 3. Luxury : Rs. 1500\n"

<< "\n Enter choice : ";

cin >> choice;

switch (choice)

{

case 1: strcpy\_s(p.room\_type, "Deluxe");

cout << "\n You chose DELUXE room.\n ";

break;

case 2: strcpy\_s(p.room\_type, "Suite ");

cout << "\n You chose SUITE room.\n ";

break;

case 3: strcpy\_s(p.room\_type, "Luxury");

cout << "\n You chose LUXURY room.\n ";

break;

default: cout << "\n Wrong choice!!! Enter a valid option...\n ";

break;

}

} while (choice != 1 && choice != 2 && choice != 3);

strcpy\_s(p.c\_name, tempnm);

p.a\_date = tempdt;

p.a\_month = tempmon;

p.a\_year = tempyr;

p.c\_no = tell + 1;

p.room\_no = tell + 1;

//Writing data in class datatypes after accepting all true value

f1.write((char\*)&p, sizeof(p)); //Appending data on file

f1.close(); //Closing link

system("pause");

}

void customer::cust\_mod() //Function for modifying customer data

{

system("cls");

fstream f1;

int no, found = 0, valid, tempdt, tempmon, tempyr, choice;

char tempnm[21];

f1.open("cust.txt", ios::binary | ios::in | ios::out);

cout << "\n\tModifying customer data...\n"

<< " Enter cust no. whose data is to be modified : ";

cin >> no;

long pos;

while (!f1.eof())

{

pos = f1.tellg();

f1.read((char\*)&p, sizeof(p)); //reading data

if (no == p.c\_no) //comparison of cust number

{

cin.get();

cout << "\n Enter new record:\n";

cout << "\n Name : ";

gets\_s(tempnm, 20);

do //Checking incorrect entry

{

valid = 0;

cout << " Year of arrival (after 1980) : ";

cin >> tempyr;

if (tempyr < 1980 || tempyr > 2016)

{

cout << "\n Wrong entry!!! Enter a valid year.\n ";

system("pause");

++valid;

}

} while (valid != 0);

do //Checking incorrect entry

{

valid = 0;

cout << " Month of arrival : ";

cin >> tempmon;

if (tempmon < 1 || tempmon > 12)

{

cout << "\n Wrong entry!!! Enter a valid month.\n ";

system("pause");

++valid;

}

} while (valid != 0);

do //Checking incorrect entry

{

valid = 0;

cout << " Date of arrival : ";

cin >> tempdt;

if (tempdt < 1 || tempmon == 2 && (((tempyr % 4 == 0) &&

(tempdt>29) || ((tempyr % 4 != 0) && tempdt > 28)) ||

tempdt> 31)

{

cout << "\n Wrong entry.... Enter a valid date.\n ";

system("pause");

++valid;

}

} while (valid != 0);

Do //Accepting room type

{

cout << "\n ROOM TYPES : "

<< "\n 1. Deluxe : Rs. 500"

<< "\n 2. Suite : Rs. 1000"

<< "\n 3. Luxury : Rs. 1500\n"

<< "\n Enter choice : ";

cin >> choice;

switch (choice)

{

case 1: strcpy\_s(p.room\_type, "Deluxe");

cout << "\n You chose DELUXE room.\n ";

break;

case 2: strcpy\_s(p.room\_type, "Suite ");

cout << "\n You chose SUITE room.\n ";

break;

case 3: strcpy\_s(p.room\_type, "Luxury");

cout << "\n You chose LUXURY room.\n ";

break;

default: cout<< "\n Wrong choice!!! Enter a valid option...\n";

break;

}

} while (choice != 1 && choice != 2 && choice != 3);

strcpy\_s(p.c\_name, tempnm);

p.a\_date = tempdt;

p.a\_month = tempmon;

p.a\_year = tempyr;

f1.seekg(pos);

f1.write((char \*)&p, sizeof(p)); //Writing modified data on file

found = 1; //flag for successful modification

cout << "\n Data successfully modified.\n ";

break;

} //End of if

} //End of while

if (found == 0)

cout << "\n Customer number not found!!!\n ";

f1.close(); //Closing link

system("pause");

} //End of customer mod function

void customer::cust\_del() //Function to delete particular customer data

{

ifstream fin;

ofstream fout;

int no, ctr = 0;

cout << "\n Enter customer number to delete : ";

cin >> no;

fin.open("cust.txt", ios::binary);

fout.open("temp.txt", ios::binary); //File for writing temporary data

while (fin.read((char \*)&p, sizeof(p))) //Reading

{

if (no == p.c\_no) //Checking for match

{

ctr = 1;

cout << "\n Record successfully deleted.\n ";

//If cust no. matches ,it is not written on temp

}

else

fout.write((char \*)&p, sizeof(p));

}

fin.close();

fout.close();

remove("cust.txt"); //Removing old file

rename("temp.txt", "cust.txt"); //Renaming temp file to cust file

if (ctr == 0)

cout << "\n Employee number not found!!!\n ";

system("pause");

} //End of customer delete function

void customer::cust\_adel() //Function to delete all customer data

{

int choice;

do

{

system("cls");

cout << " Are you sure you want to delete all data???\n"

<< "\n 1. Yes"

<< "\n 2. No\n"

<< "\n Enter choice : ";

cin >> choice;

switch (choice)

{

case 1: remove("cust.txt"); //Removing file

f.open("cust.txt", ios::out | ios::binary | ios::in);

p.c\_no = 0;

p.room\_no = 0;

f.close();

cout << "\n Data successfully deleted.\n ";

system("pause");

break;

case 2: break;

default: cout << "\n Wrong choice!!! Enter a valid option...\n";

break;

}

} while (choice != 1 && choice != 2);

} //End of function

void customer::cust\_disp() //Function to display customer data

{

fstream f1;

system("cls");

cout <<"\n\t\t\t\t\t Customer Details\n"

<< " Customer No. Name Room No. Room Type Arrival

Departure\n";

int c = 0;

f1.open("cust.txt", ios::in | ios::binary);

f1.seekg(0, ios::beg); //Bring file pointer to beginning

while (f1.read((char\*)&p, sizeof(p))) //Read from beginning

{

cout << "\n " << p.c\_no << " "

<< p.c\_name << "\t "

<< p.room\_no << " "

<< p.room\_type << " "

<< p.a\_date << " / " << p.a\_month << " / " << p.a\_year << " ";

if (p.d\_date == 0)

cout << " No data available";

else

cout << " " << p.d\_date << " / " << p.d\_month << " / " << p.d\_year;

}

cout << "\n\n ";

system("pause");

f1.close();

} //End of display function

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Class-3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class controlling all the classes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class control

{

private:

int ch;

public:

customer cust;

food d;

void pass() //Function for obtaining password to login

{

string pass;

char ch;

do

{

system("cls");

cout << " \*\*\*\*\*\*\*\*\*\*\* WELCOME TO HAMILTON HOTEL \*\*\*\*\*\*\*\*\*\*\*.\n";

cout << "\n Enter password :\t";

ch = \_getch();

while (ch != 13)

//accepting password without displaying characters on screen

{

pass.push\_back(ch);

cout << '\*';

ch = \_getch();

}

if (pass=="admin")

{

cout << "\n\n Password entered successfully.\n ";

pass.clear();

system("pause");

break; //

}

else

{

cout << "\n\n Wrong password!!! Enter correct password.\n ";

pass.clear();

system("pause");

}

} while (pass != "admin");

mmenu();

} //End of pass function

void mmenu() //Function for displaying main menu

{

int choice;

do

{

system("cls");

cout << "\*\*\*\*\*\*\*\*\*\*\*\* HOTEL MANAGEMENT \*\*\*\*\*\*\*\*\*\*\*\*"

<< "\n MAIN MENU\n"

<< "\n 1. Information"

<< "\n 2. Customer Details"

<< "\n 3. Food Details"

<< "\n 4. Customer Billing"

<< "\n 5. Exit\n"

<< "\n Enter choice : ";

cin >> choice;

switch (choice)

{

case 1: information();

break;

case 2: cust.cust\_detail();

break;

case 3: d.food\_menu();

break;

case 4: cust.cust\_bill();

break;

case 5: break;

default: cout << "Wrong choice!!! Enter a valid option...\n";

system("pause");

break;

}

} while (choice != 5);

exit(0);

} //End of main menu

void information() //Function for displaying information

{

system("cls");

cout << "\n\t\t\t\t\tWELCOME TO HAMILTON HOTEL\n\n"

<< " ABOUT HAMILTON\n"

<< "\n HAMILTON exposes the creative and curious traveller to the most

unexpected and engaging experiences"

<< "\n their destinaion has to offer. Through curated cultural

moments, both inside and outside the hotel,"

<< "\n guests leave feeling fulfilled, having made a unique discovery

they can share all their next stop.\n\n"

<< "\n HISTORY\n"

<< "\n 1972 : Air France creates Hamilton as a 'home away from home'

for its travellers.\n"

<< "\n 1979 : Hamilton expands to 21 hotels in destinations across

Europe, Africa,"

<< "\n Canada, South America, the Middle East and Mauritius.\n"

<< "\n 1991 : Total number of hotel rises to 58!\n"

<< "\n 2005 : Hamilton joins Starwood Hotels & Resort Worldwide Inc.

as one of its premier lifestyle brands.\n"

<< "\n 2012 : Hamilton celebrates 40 years of excellence in travel.\n"

<< "\n 2013 : With a mission to engage in its most unexpected

experiences their destination has to offer,"

<< "\n Hamilton repositions itself, and refreshes its brand

identity. And also opens 11 new hotels,"

<< "\n nearly tripling its global footprint.\n"

<< "\n 2015 : Hamilton continues to grow and bring its guests inspired

experiences in over 100 hotels and"

<< "\n resorts in 35 countries around the globe.\n\n"

<< "\n FOR ANY QUERIES , please contact our Customer Care :\n\n"

<< " M. No.: +91 9988223567\t\t\t\t Telephone: +91 1127639386\n"

<< " Email:cust\_care@hamiltonhotelgroups.com\t Website:

www.hamiltonhotelgroups.com\n\n ";

system("pause");

}

}; //End of class control

void main() //Main function

{

system("cls");

control c;

c.pass();

} //End of main function

# **Output**









 



 

 



 

 

 

 



 

 







 

