AIR UNIVERSITY

# Hotel Management System

Object-Oriented Programming

**Group Members:**

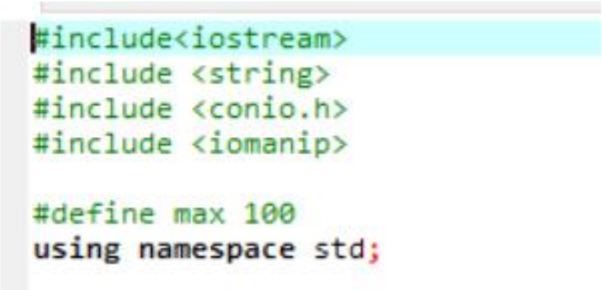
| | |
|---|---|
| • Mubahil Ahmad | (211037) |
| • Ahsan Ahmed | (211059) |
| • Syed Ali Zain Ul Abedin Naqvi | (211113) |
| • Hadia Abbas | (211047) |
| • Salwa Shahid Farooqi | (211063) |

# Introduction

This project is all about managing the basic needs of a hotel. Which cover the scope from checking in to checking out including managing rooms and summary of customers. There are three classes which are interconnected to each other. Two have composition/ aggregation and one is inherited. A static data member is used to calculate the total number of rooms. The program is also password protected and will terminate after multiple wrong attempts. You can also manage the details of room such as rent and other details like air conditioning   the following are the details of functions of the program:

# Header files:

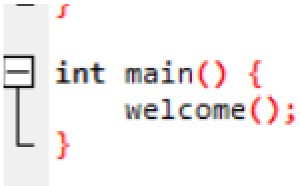
We are using following header files



```
#include<iostream>
#include <string>
#include <conio.h>
#include <iomanip>

#define max 100
using namespace std;
```

# The main function:

There is only one function in the main function called welcome()

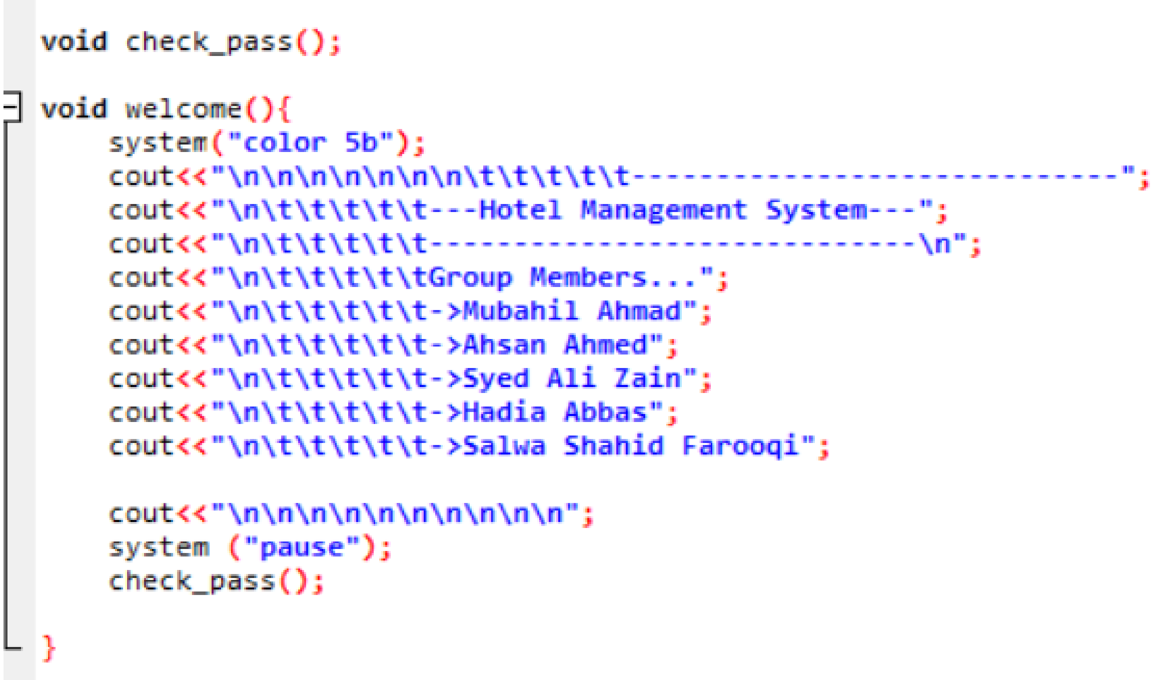


```
}

int main() {
    welcome();
}
```

# Welcome():

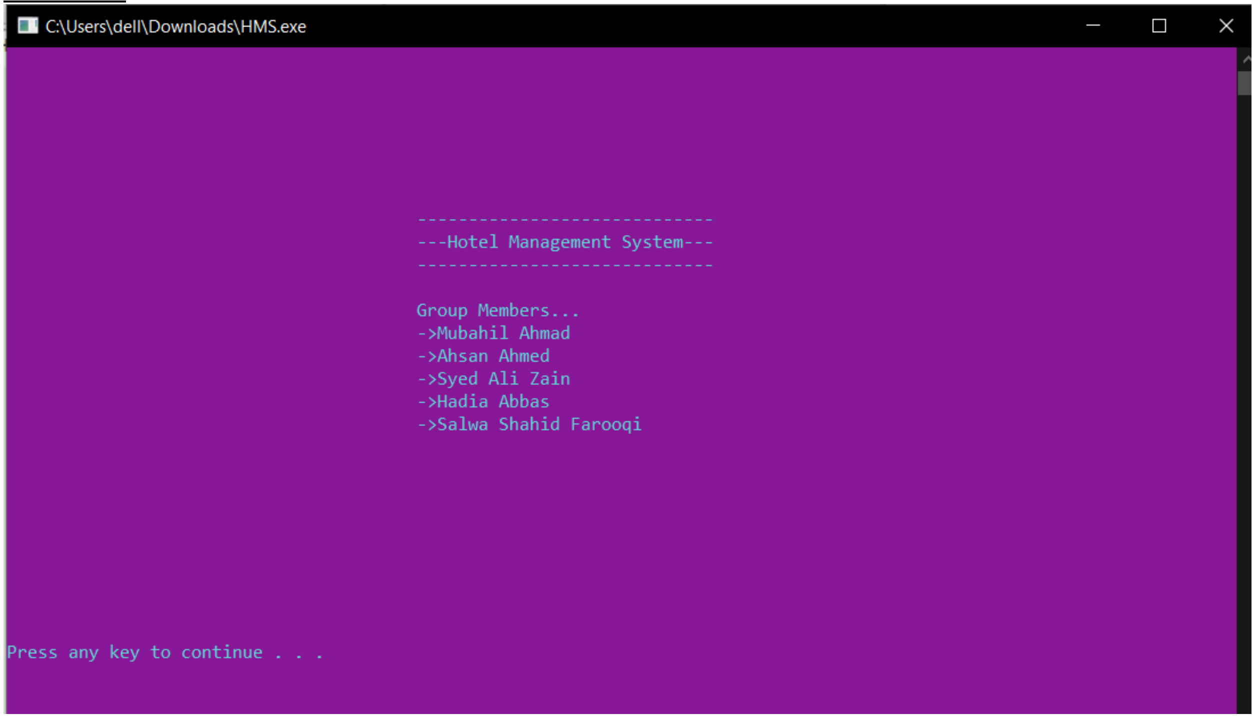
We use the system ("pause") command which is way to pause a program and wait for a keyboard input to continue so that the program starts when we want it to once we press any key the check_pass() function will be called First, we are using the system("color 5b") command to change the background color of console to purple and the text color to light aqua.

Code



```
void check_pass();

void welcome(){
	system("color 5b");
	cout<<"\n\n\n\n\n\n\n\t\t\t\t\t-----------------------------";
	cout<<"\n\t\t\t\t\t---Hotel Management System---";
	cout<<"\n\t\t\t\t\t-----------------------------\n";
	cout<<"\n\t\t\t\t\t\tGroup Members...";
	cout<<"\n\t\t\t\t\t\t->Mubahil Ahmad";
	cout<<"\n\t\t\t\t\t\t->Ahsan Ahmed";
	cout<<"\n\t\t\t\t\t\t->Syed Ali Zain";
	cout<<"\n\t\t\t\t\t\t->Hadia Abbas";
	cout<<"\n\t\t\t\t\t\t->Salwa Shahid Farooqi";

	cout<<"\n\n\n\n\n\n\n\n\n\n";
	system ("pause");
	check_pass();

}
```

Console



```
                                        -----------------------------
                                        ---Hotel Management System---
                                        -----------------------------

                                        Group Members...
                                        ->Mubahil Ahmad
                                        ->Ahsan Ahmed
                                        ->Syed Ali Zain
                                        ->Hadia Abbas
                                        ->Salwa Shahid Farooqi









Press any key to continue . . .
```

## Check_pass:

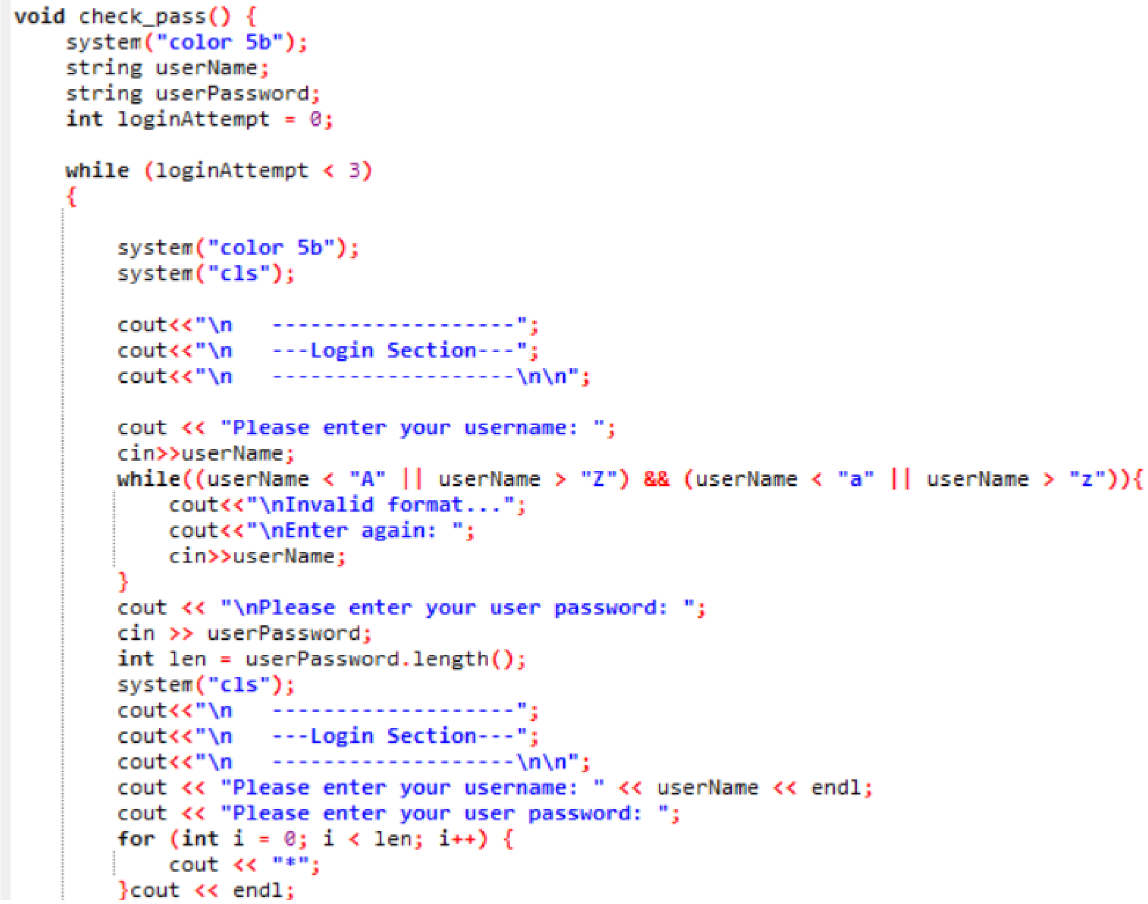
We use system("cls") which is used to make the screen/terminal clear

We have declared to strings username and user password to verify username and password we have also created a variable int login attempt and initialized it as zero

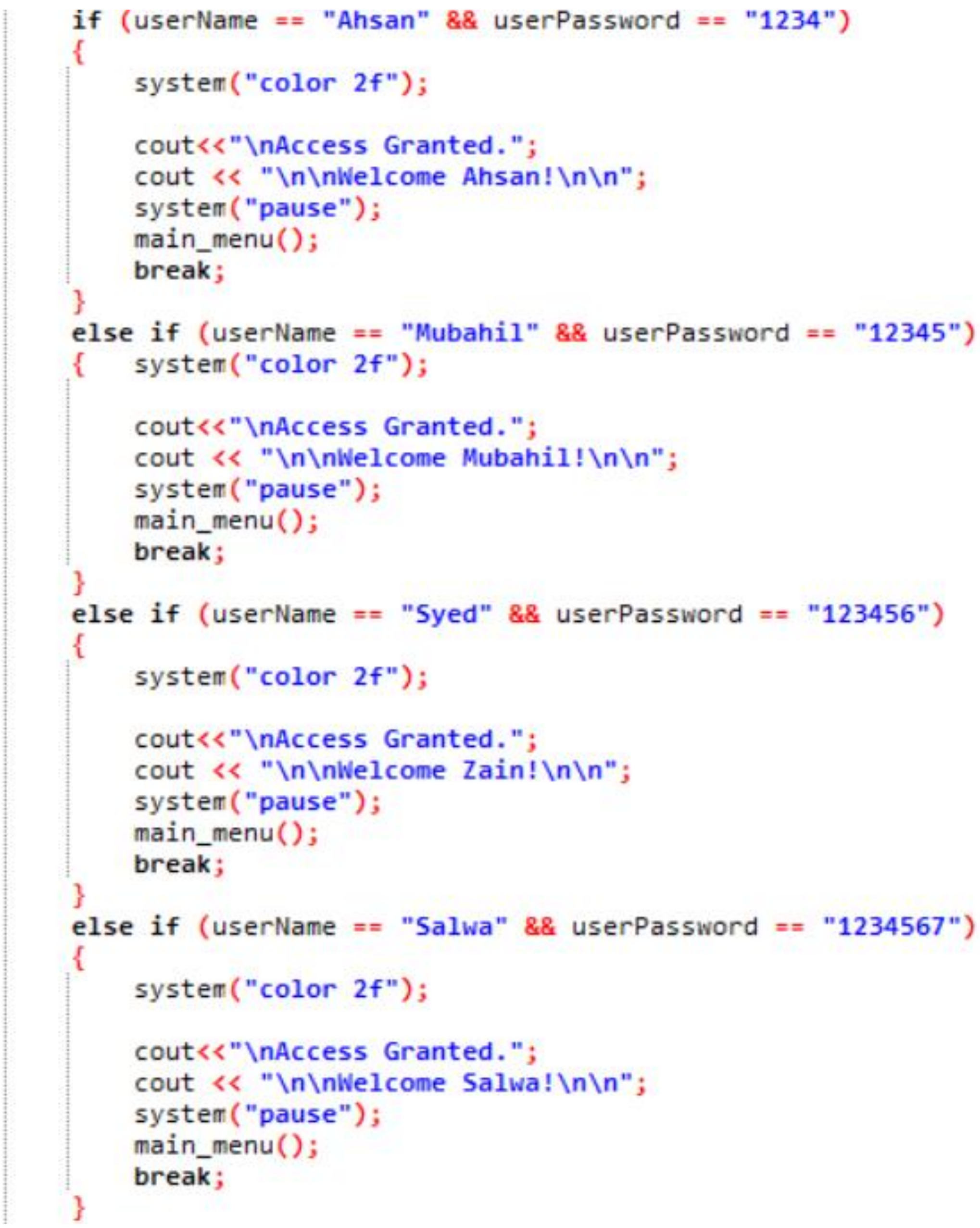
The password input and verifying part will only work as long as the login attempts are less than 3 after 3 wrong attempts the program will terminate  we check the password and username using if else statements in which have entered 5 users and gave each of them their unique password and the program will only run if both the username and password are correct if the the background will change to green and welcome screen will appear and you will be sent to the main_menu() function and the if the entered credentials are wrong the background color will change to red and the system will tell you that the entered username or password are incorrect and will ask for the credentials again this will continue while the login attempts are less than 3

We have also used the .length() function which calculates the length of the function and using for loop hides the password with steric

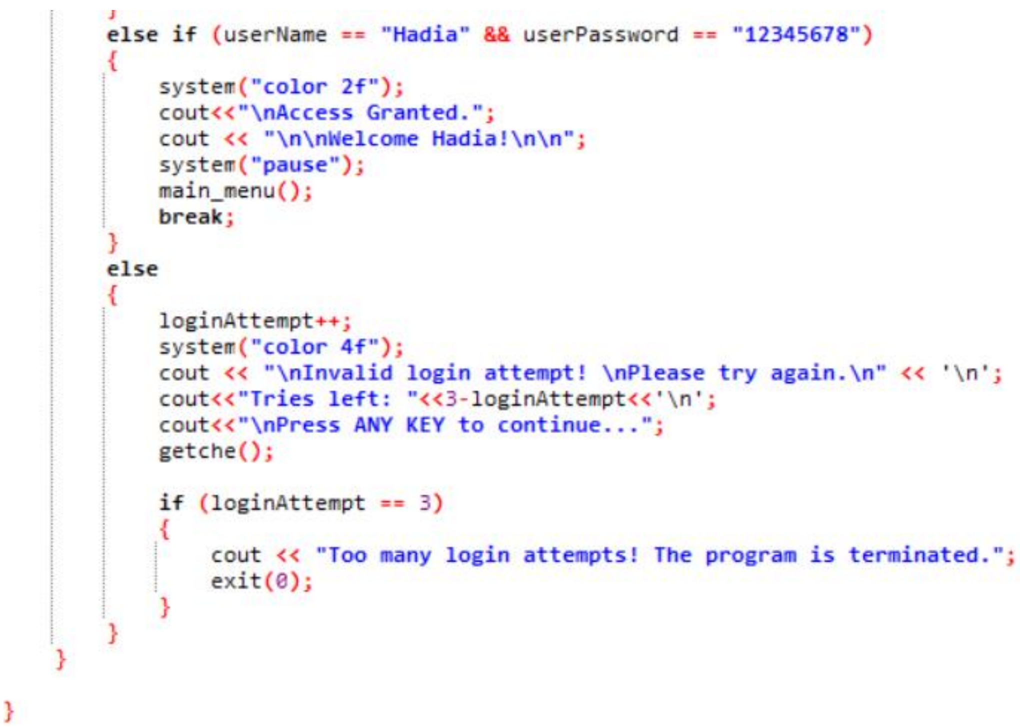
Code



```
void check_pass() {
    system("color 5b");
    string userName;
    string userPassword;
    int loginAttempt = 0;

    while (loginAttempt < 3)
    {

        system("color 5b");
        system("cls");

        cout<<"\n   -------------------";
        cout<<"\n   ---Login Section---";
        cout<<"\n   -------------------\n\n";

        cout << "Please enter your username: ";
        cin>>userName;
        while((userName < "A" || userName > "Z") && (userName < "a" || userName > "z")){
            cout<<"\nInvalid format...";
            cout<<"\nEnter again: ";
            cin>>userName;
        }
        cout << "\nPlease enter your user password: ";
        cin >> userPassword;
        int len = userPassword.length();
        system("cls");
        cout<<"\n   -------------------";
        cout<<"\n   ---Login Section---";
        cout<<"\n   -------------------\n\n";
        cout << "Please enter your username: " << userName << endl;
        cout << "Please enter your user password: ";
        for (int i = 0; i < len; i++) {
            cout << "*";
        }cout << endl;
```

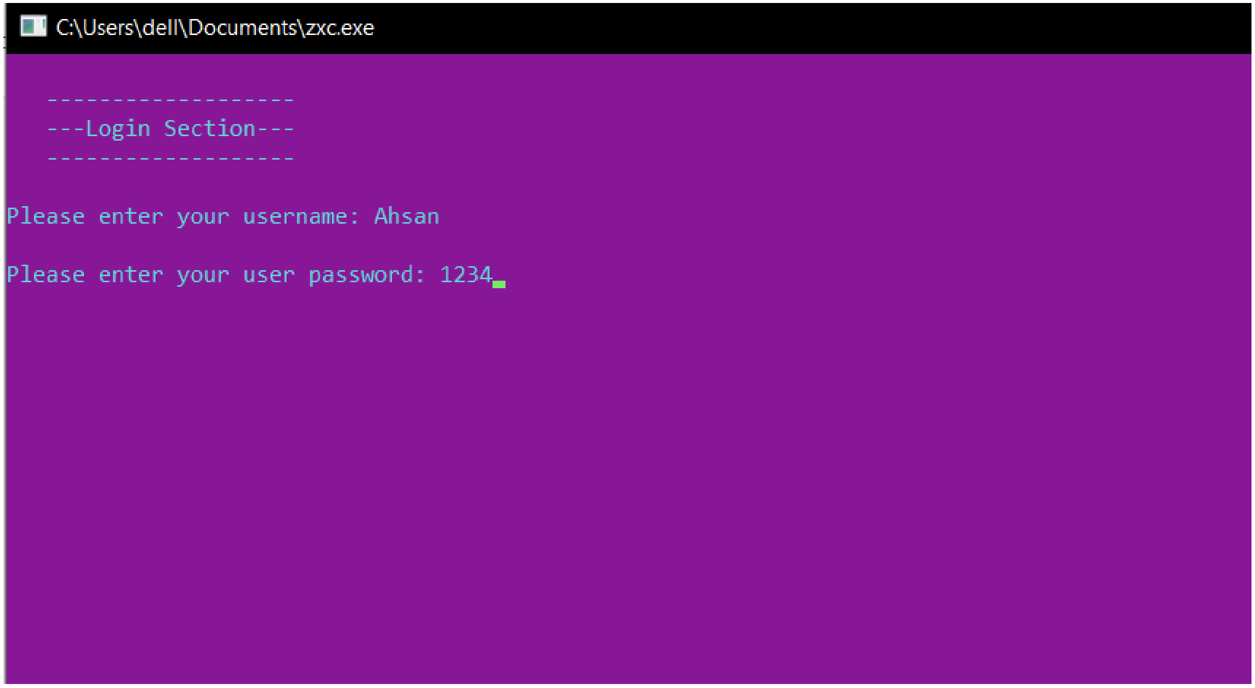


```
if (userName == "Ahsan" && userPassword == "1234")
{
    system("color 2f");

    cout<<"\nAccess Granted.";
    cout << "\n\nWelcome Ahsan!\n\n";
    system("pause");
    main_menu();
    break;
}
else if (userName == "Mubahil" && userPassword == "12345")
{   system("color 2f");

    cout<<"\nAccess Granted.";
    cout << "\n\nWelcome Mubahil!\n\n";
    system("pause");
    main_menu();
    break;
}
else if (userName == "Syed" && userPassword == "123456")
{
    system("color 2f");

    cout<<"\nAccess Granted.";
    cout << "\n\nWelcome Zain!\n\n";
    system("pause");
    main_menu();
    break;
}
else if (userName == "Salwa" && userPassword == "1234567")
{
    system("color 2f");

    cout<<"\nAccess Granted.";
    cout << "\n\nWelcome Salwa!\n\n";
    system("pause");
    main_menu();
    break;
}
```



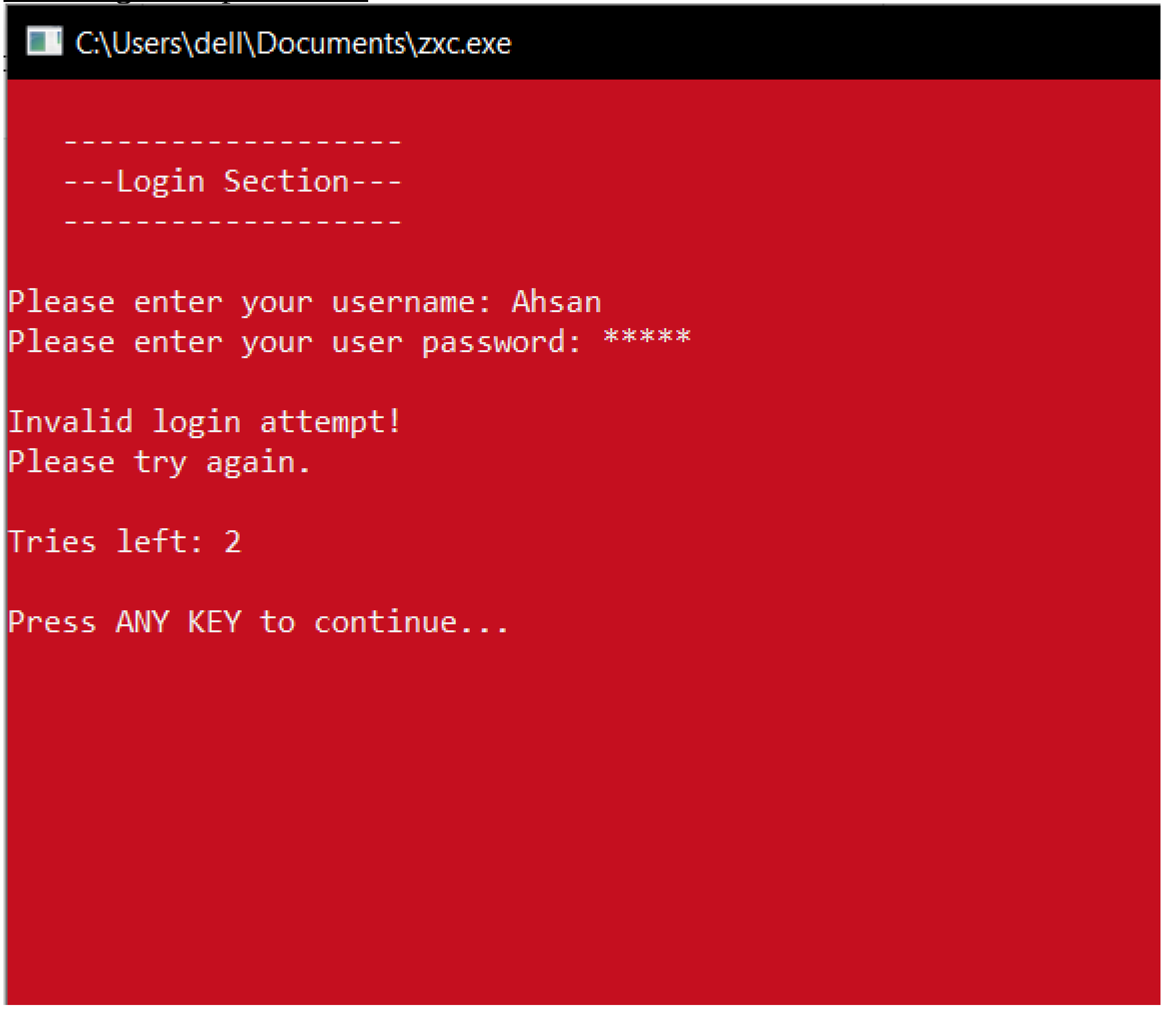
```
        }
        else if (userName == "Hadia" && userPassword == "12345678")
        {
            system("color 2f");
            cout<<"\nAccess Granted.";
            cout << "\n\nWelcome Hadia!\n\n";
            system("pause");
            main_menu();
            break;
        }
        else
        {
            loginAttempt++;
            system("color 4f");
            cout << "\nInvalid login attempt! \nPlease try again.\n" << '\n';
            cout<<"Tries left: "<<3-loginAttempt<<'\n';
            cout<<"\nPress ANY KEY to continue...";
            getche();

            if (loginAttempt == 3)
            {
                cout << "Too many login attempts! The program is terminated.";
                exit(0);
            }
        }
    }

}
```

Console:



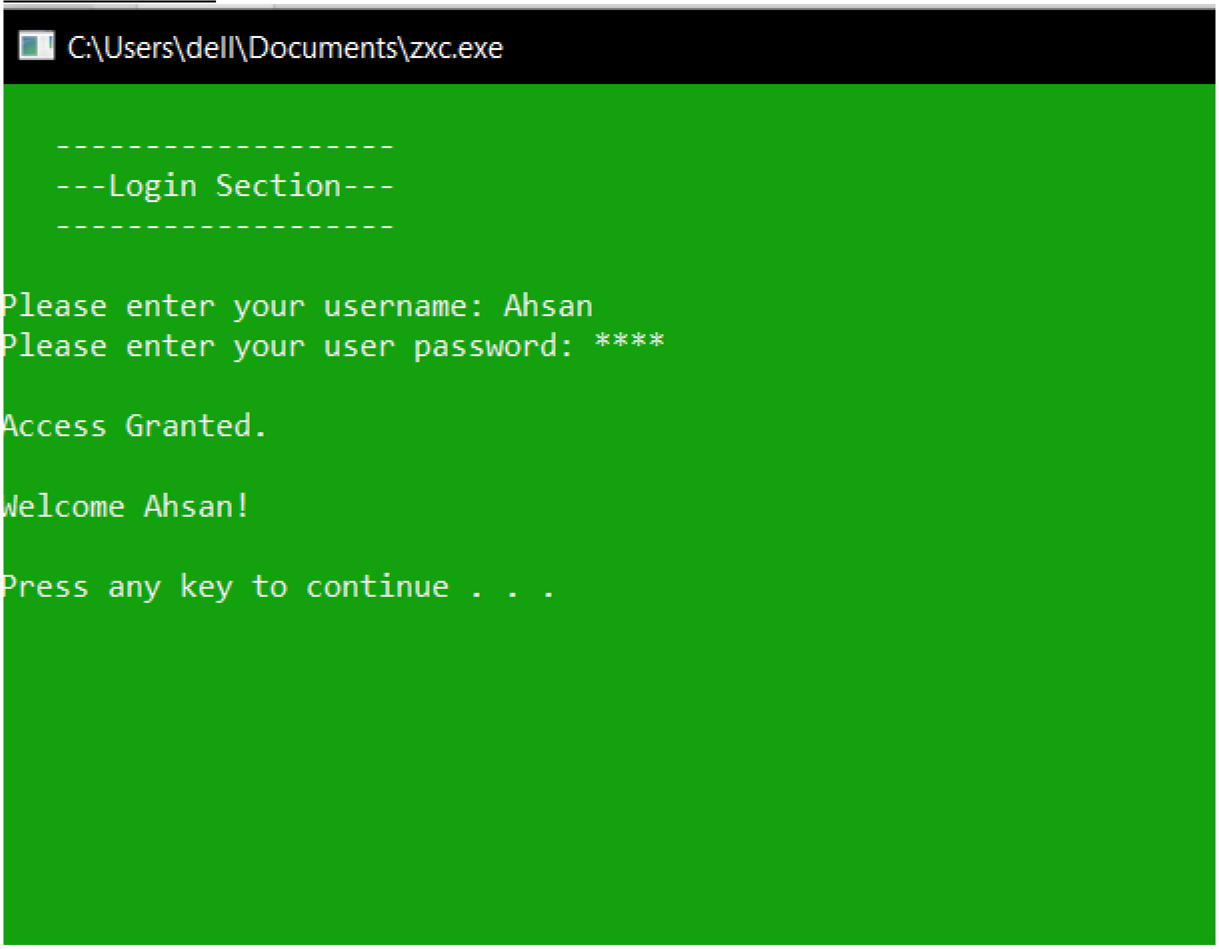
```
   -------------------
   ---Login Section---
   -------------------

Please enter your username: Ahsan

Please enter your user password: 1234_
```

If wrong multiple times:



```
   -------------------
   ---Login Section---
   -------------------

Please enter your username: Ahsan
Please enter your user password: *****

Invalid login attempt!
Please try again.

Tries left: 2

Press ANY KEY to continue...
```

When correct:

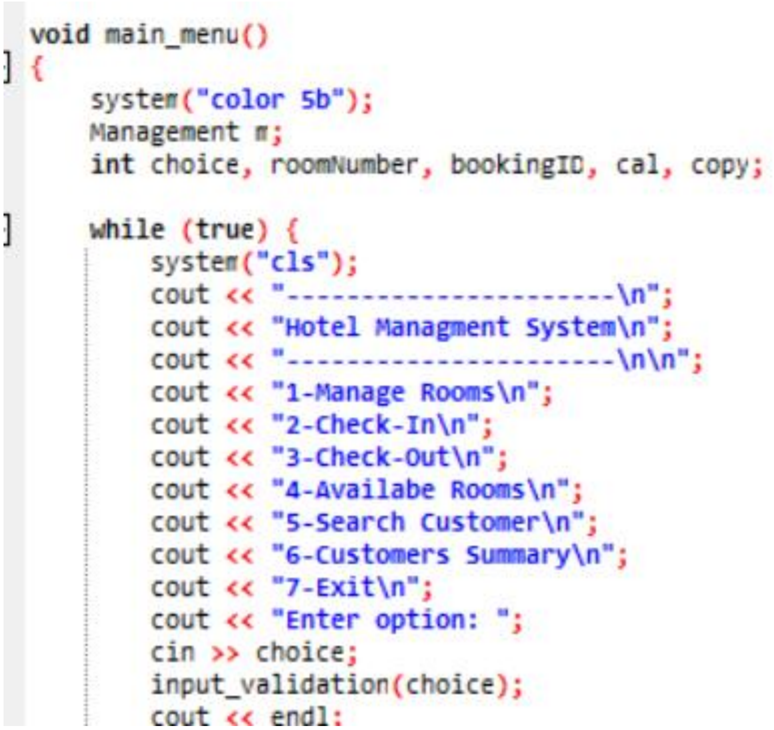


```
   -------------------
   ---Login Section---
   -------------------

Please enter your username: Ahsan
Please enter your user password: ****

Access Granted.

Welcome Ahsan!

Press any key to continue . . .
```

**Main_menu():**

In the main menu function we have created an object management m and initialized 5 variables int choice, roomNumber, bookingID, cal, and copy

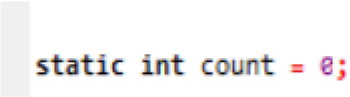
We are using while loop to validate that one the given options is selected if any other option is selected it will show enter correct option once we select an option we will be sent to relevant function



```
void main_menu()
{
    system("color 5b");
    Management m;
    int choice, roomNumber, bookingID, cal, copy;

    while (true) {
        system("cls");
        cout << "----------------------\n";
        cout << "Hotel Managment System\n";
        cout << "----------------------\n\n";
        cout << "1-Manage Rooms\n";
        cout << "2-Check-In\n";
        cout << "3-Check-Out\n";
        cout << "4-Availabe Rooms\n";
        cout << "5-Search Customer\n";
        cout << "6-Customers Summary\n";
        cout << "7-Exit\n";
        cout << "Enter option: ";
        cin >> choice;
        input_validation(choice);
        cout << endl;
```

## Room counter:

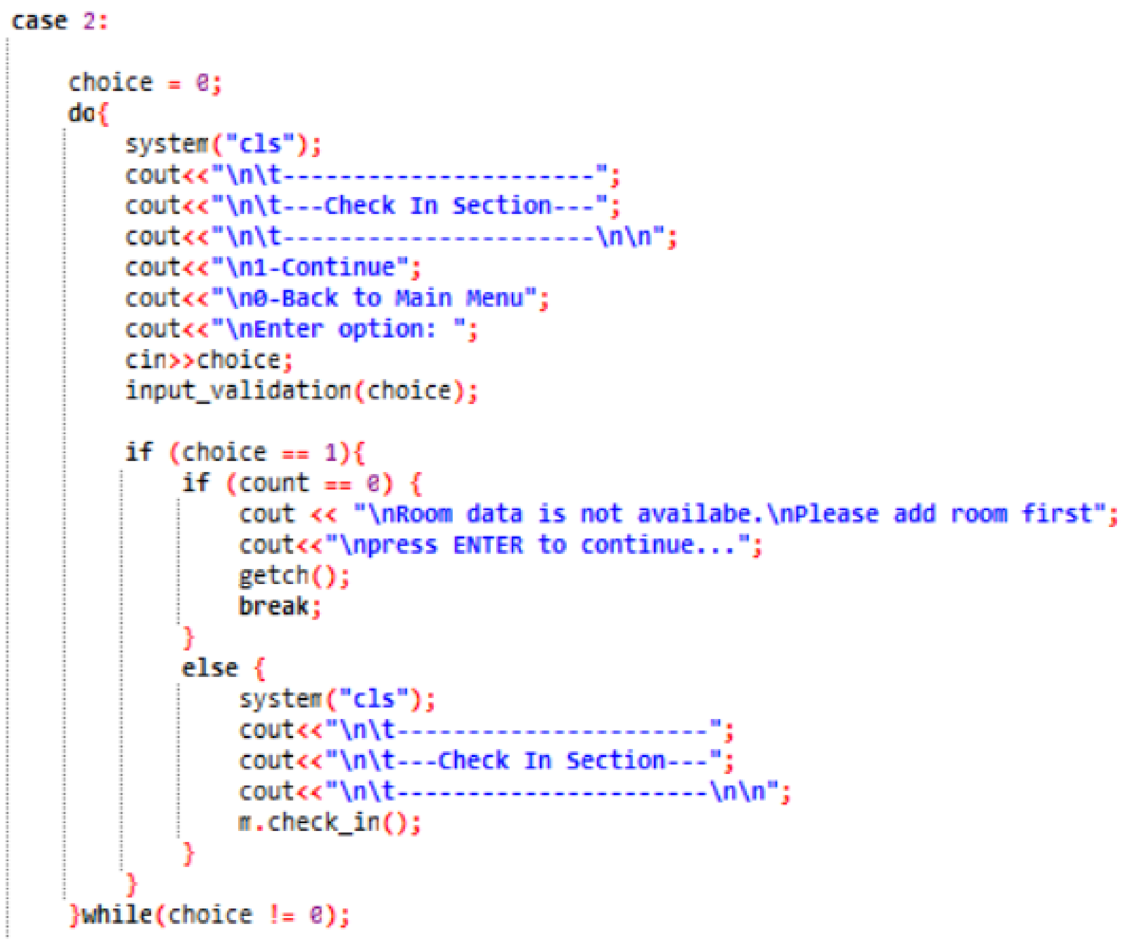
We are counting the added room using static int count we will show the process to add room later.



```
static int count = 0;
```

## Check In:

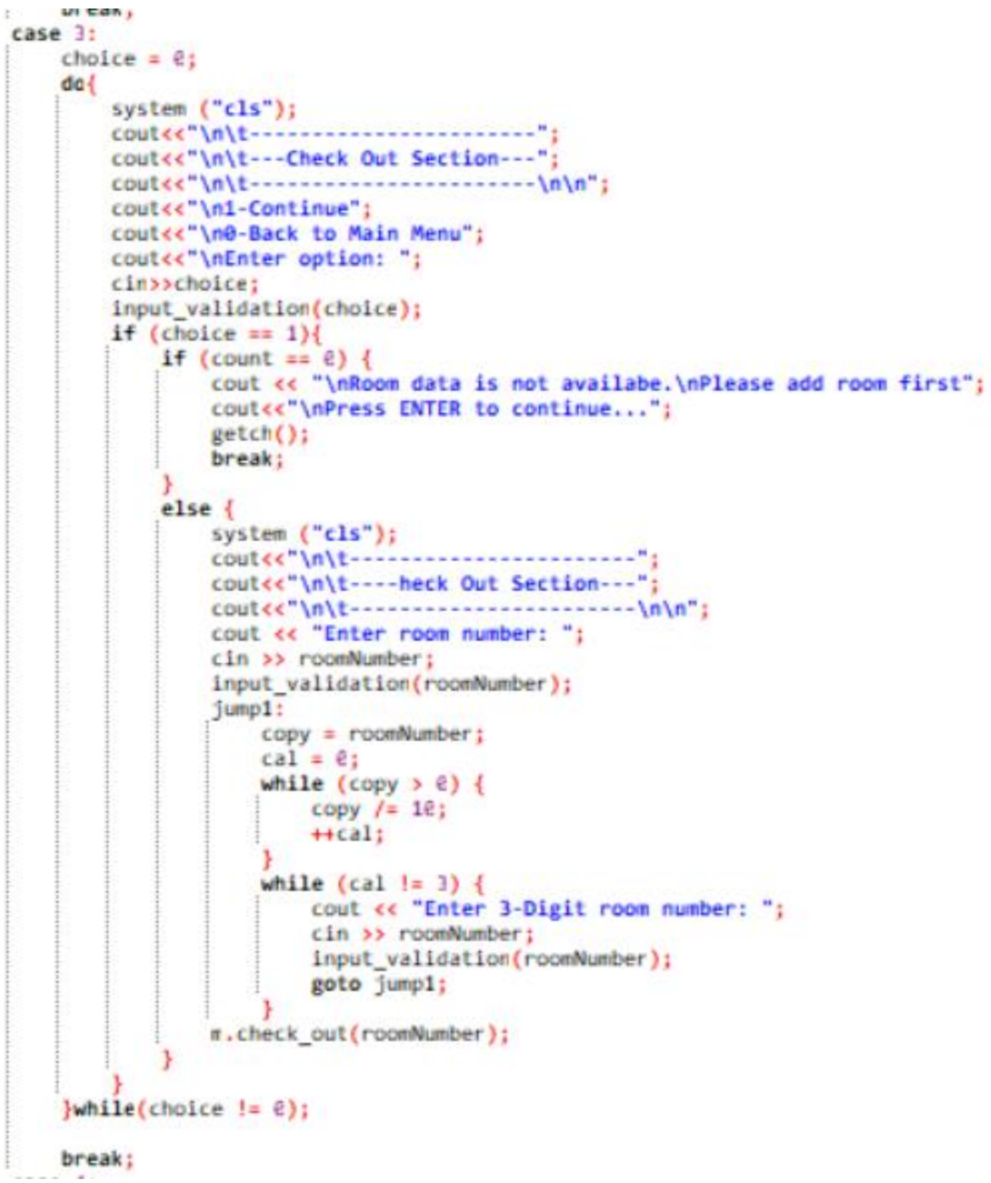
When we check in it will check whether or not if we have added a room or not if we have not added a room and count=0 then it will show to room data is not available and to add a room first and if room is available it will ask for customer details



```
case 2:

    choice = 0;
    do{
        system("cls");
        cout<<"\n\t----------------------";
        cout<<"\n\t---Check In Section---";
        cout<<"\n\t----------------------\n\n";
        cout<<"\n1-Continue";
        cout<<"\n0-Back to Main Menu";
        cout<<"\nEnter option: ";
        cin>>choice;
        input_validation(choice);

        if (choice == 1){
            if (count == 0) {
                cout << "\nRoom data is not availabe.\nPlease add room first";
                cout<<"\npress ENTER to continue...";
                getch();
                break;
            }
            else {
                system("cls");
                cout<<"\n\t----------------------";
                cout<<"\n\t---Check In Section---";
                cout<<"\n\t----------------------\n\n";
                m.check_in();
            }
        }
    }while(choice != 0);
```

## Check out:

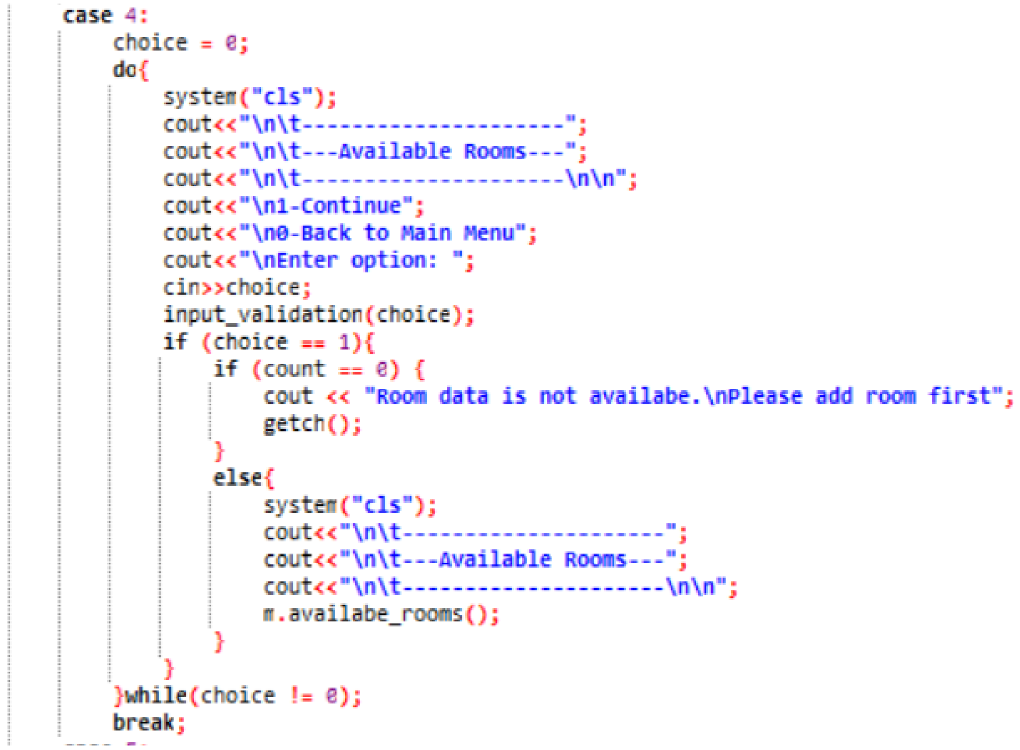
The check out function will display the total bill after deducting the advance payment from their bill and the room will be once again available for booking



```
    break;
case 3:
    choice = 0;
    do{
        system ("cls");
        cout<<"\n\t-----------------------";
        cout<<"\n\t---Check Out Section---";
        cout<<"\n\t-----------------------\n\n";
        cout<<"\n1-Continue";
        cout<<"\n0-Back to Main Menu";
        cout<<"\nEnter option: ";
        cin>>choice;
        input_validation(choice);
        if (choice == 1){
            if (count == 0) {
                cout << "\nRoom data is not availabe.\nPlease add room first";
                cout<<"\nPress ENTER to continue...";
                getch();
                break;
            }
            else {
                system ("cls");
                cout<<"\n\t-----------------------";
                cout<<"\n\t----heck Out Section---";
                cout<<"\n\t-----------------------\n\n";
                cout << "Enter room number: ";
                cin >> roomNumber;
                input_validation(roomNumber);
                jump1:
                    copy = roomNumber;
                    cal = 0;
                    while (copy > 0) {
                        copy /= 10;
                        ++cal;
                    }
                    while (cal != 3) {
                        cout << "Enter 3-Digit room number: ";
                        cin >> roomNumber;
                        input_validation(roomNumber);
                        goto jump1;
                    }
                r.check_out(roomNumber);
            }
        }
    }while(choice != 0);

    break;
```

## Available Rooms:

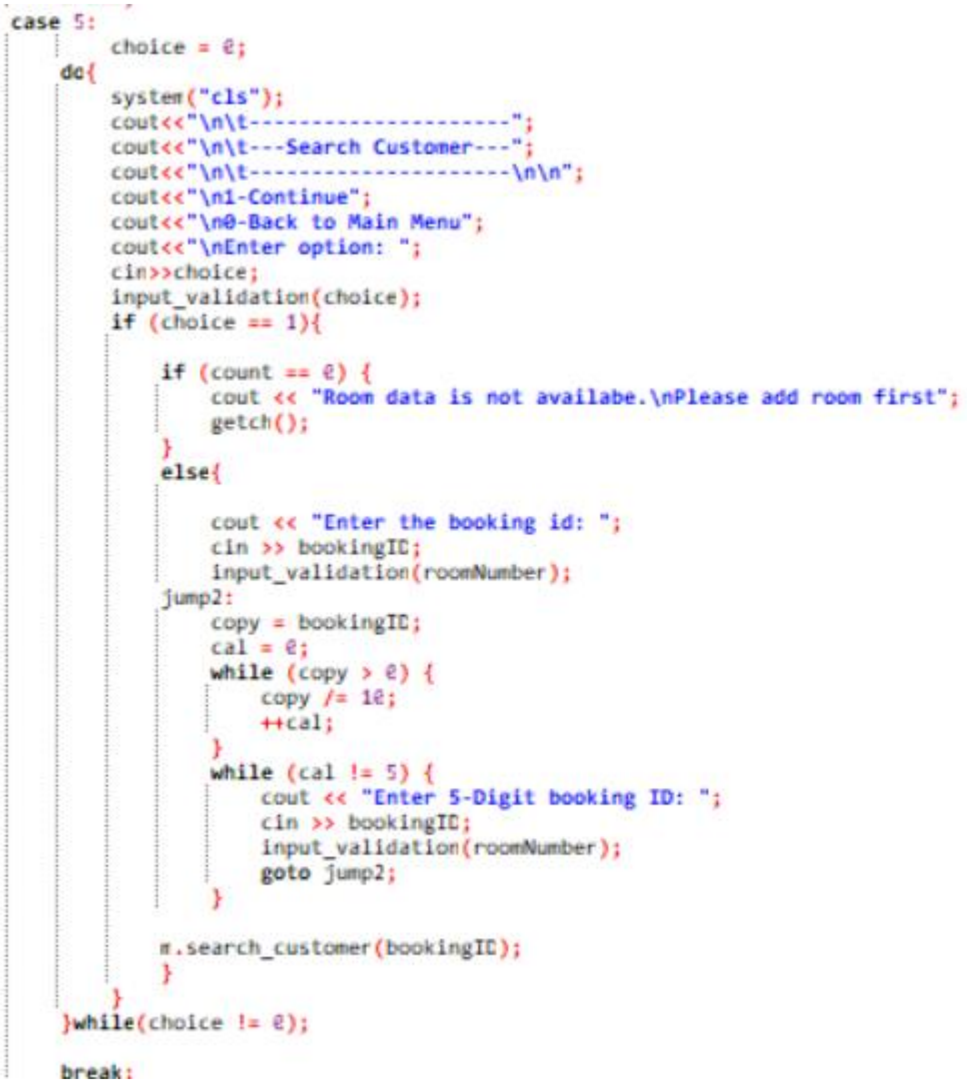
This will show you the details of all the available un occupied rooms and if no rooms are available, it will display that no room is available .



```
case 4:
    choice = 0;
    do{
        system("cls");
        cout<<"\n\t---------------------";
        cout<<"\n\t---Available Rooms---";
        cout<<"\n\t---------------------\n\n";
        cout<<"\n1-Continue";
        cout<<"\n0-Back to Main Menu";
        cout<<"\nEnter option: ";
        cin>>choice;
        input_validation(choice);
        if (choice == 1){
            if (count == 0) {
                cout << "Room data is not availabe.\nPlease add room first";
                getch();
            }
            else{
                system("cls");
                cout<<"\n\t---------------------";
                cout<<"\n\t---Available Rooms---";
                cout<<"\n\t---------------------\n\n";
                m.availabe_rooms();
            }
        }
    }while(choice != 0);
    break;
```

## Search Customer:

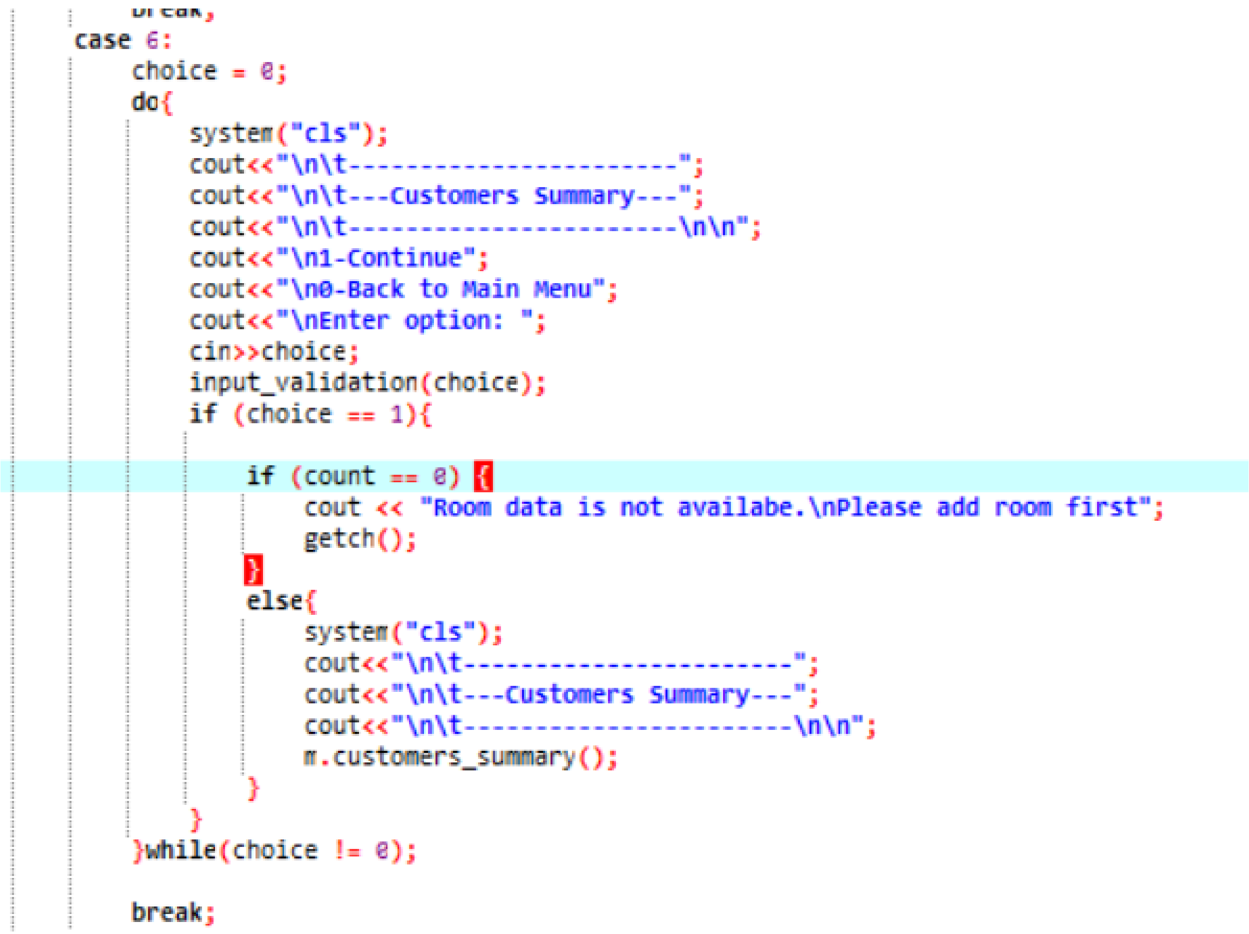
The search customer function will display the details using his booking id if person is not available it will show that person is not available it will also validate if the booking id is 5 digit or not >in case it is not 5 digit it will ask you to enter 5-digit id and then search in stored data for the customer



```
case 5:
        choice = 0;
    do{
        system("cls");
        cout<<"\n\t--------------------";
        cout<<"\n\t---Search Customer---";
        cout<<"\n\t--------------------\n\n";
        cout<<"\n1-Continue";
        cout<<"\n0-Back to Main Menu";
        cout<<"\nEnter option: ";
        cin>>choice;
        input_validation(choice);
        if (choice == 1){

            if (count == 0) {
                cout << "Room data is not availabe.\nPlease add room first";
                getch();
            }
            else{

                cout << "Enter the booking id: ";
                cin >> bookingID;
                input_validation(roomNumber);
            jump2:
                copy = bookingID;
                cal = 0;
                while (copy > 0) {
                    copy /= 10;
                    ++cal;
                }
                while (cal != 5) {
                    cout << "Enter 5-Digit booking ID: ";
                    cin >> bookingID;
                    input_validation(roomNumber);
                    goto jump2;
                }

            m.search_customer(bookingID);
            }
        }
    }while(choice != 0);

    break;
```

## Customer Summary:

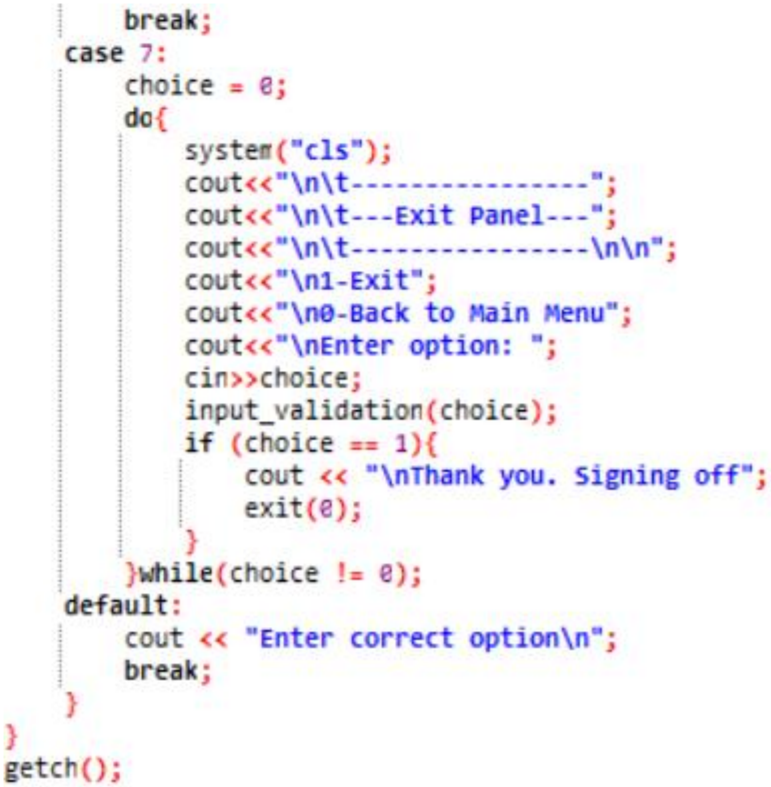
The customer summary function will display the details of all the customers staying in the hotel and will display room data is not available if no data is entered



```
        break;
    case 6:
        choice = 0;
        do{
            system("cls");
            cout<<"\n\t-----------------------";
            cout<<"\n\t---Customers Summary---";
            cout<<"\n\t-----------------------\n\n";
            cout<<"\n1-Continue";
            cout<<"\n0-Back to Main Menu";
            cout<<"\nEnter option: ";
            cin>>choice;
            input_validation(choice);
            if (choice == 1){

                if (count == 0) {
                    cout << "Room data is not availabe.\nPlease add room first";
                    getch();
                }
                else{
                    system("cls");
                    cout<<"\n\t-----------------------";
                    cout<<"\n\t---Customers Summary---";
                    cout<<"\n\t-----------------------\n\n";
                    m.customers_summary();
                }
            }
        }while(choice != 0);

        break;
```

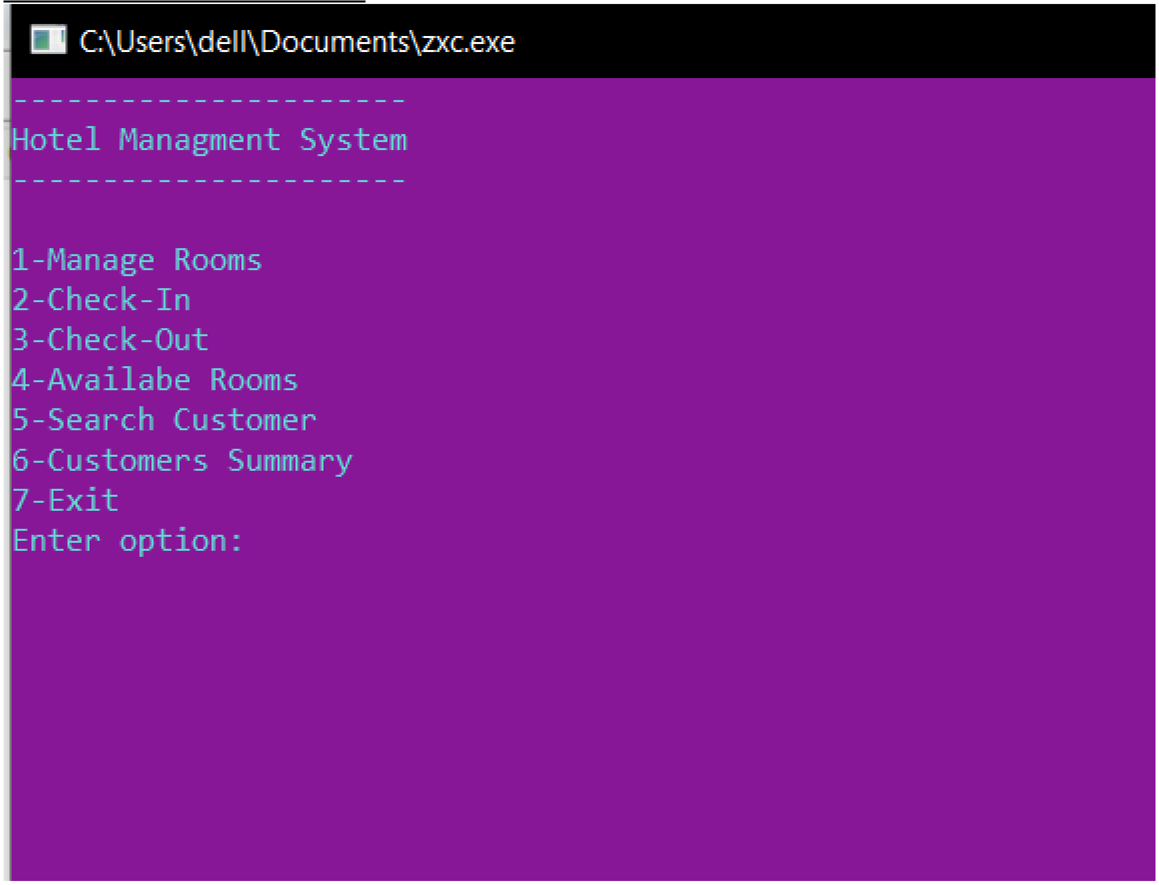
## Exit:

The exit function will terminate the program



```
        break;
    case 7:
        choice = 0;
        do{
            system("cls");
            cout<<"\n\t----------------";
            cout<<"\n\t---Exit Panel---";
            cout<<"\n\t----------------\n\n";
            cout<<"\n1-Exit";
            cout<<"\n0-Back to Main Menu";
            cout<<"\nEnter option: ";
            cin>>choice;
            input_validation(choice);
            if (choice == 1){
                cout << "\nThank you. Signing off";
                exit(0);
            }
        }while(choice != 0);
    default:
        cout << "Enter correct option\n";
        break;
    }
}
getch();
```

Console for main menu:



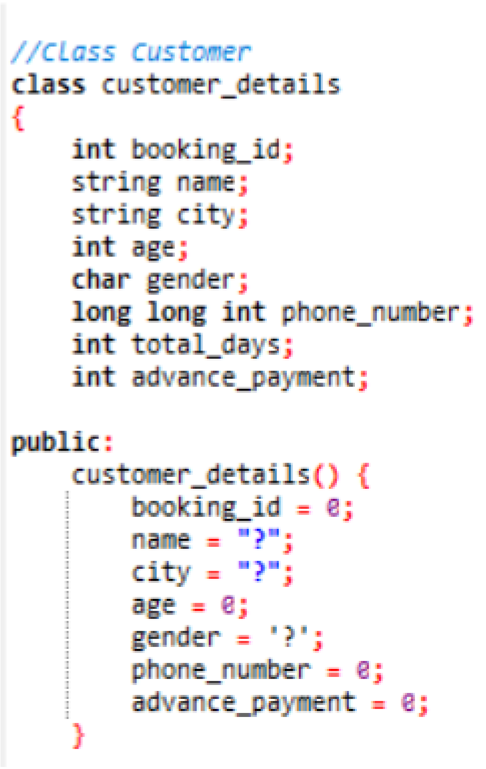
```
----------------------
Hotel Managment System
----------------------

1-Manage Rooms
2-Check-In
3-Check-Out
4-Availabe Rooms
5-Search Customer
6-Customers Summary
7-Exit
Enter option:
```

## Class Customer_details:

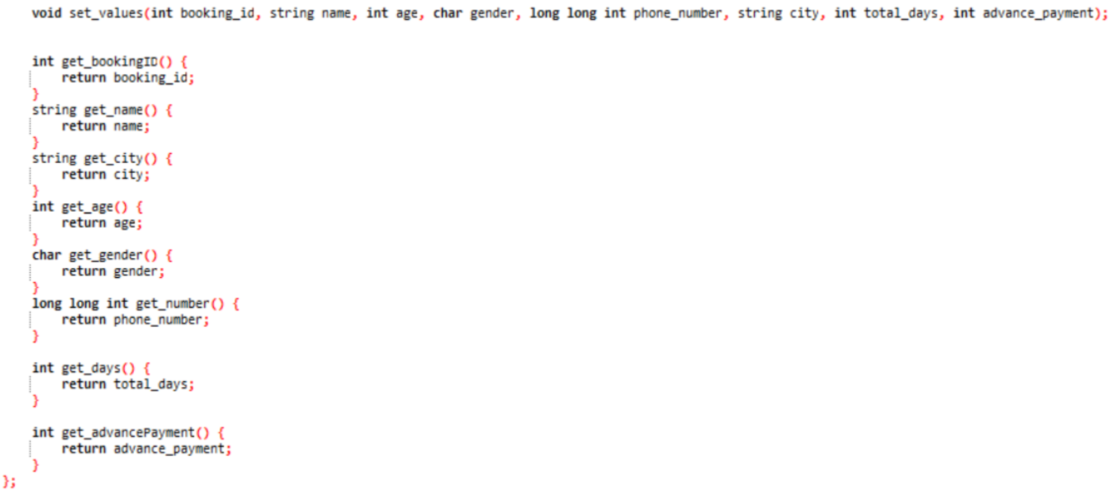
In the customer details class, we have created several variables int booking_id ,string name ,string city ,int age, char gender, long long int phone_number ,int total_days ,int advance_payment and made them public using access specifier This class contains all the variable needed to store customers personal details

### Getter and Setter:

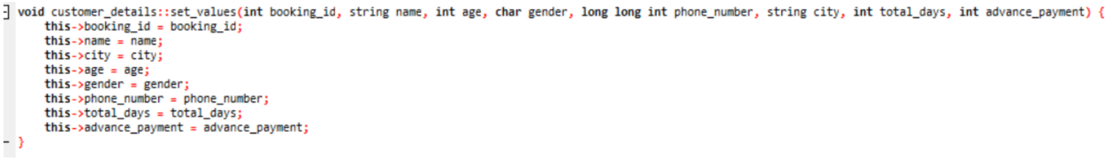
We have created getter function retrieve the variable value and setter to set the variable value



```
//Class Customer
class customer_details
{
    int booking_id;
    string name;
    string city;
    int age;
    char gender;
    long long int phone_number;
    int total_days;
    int advance_payment;

public:
    customer_details() {
        booking_id = 0;
        name = "?";
        city = "?";
        age = 0;
        gender = '?';
        phone_number = 0;
        advance_payment = 0;
    }
```



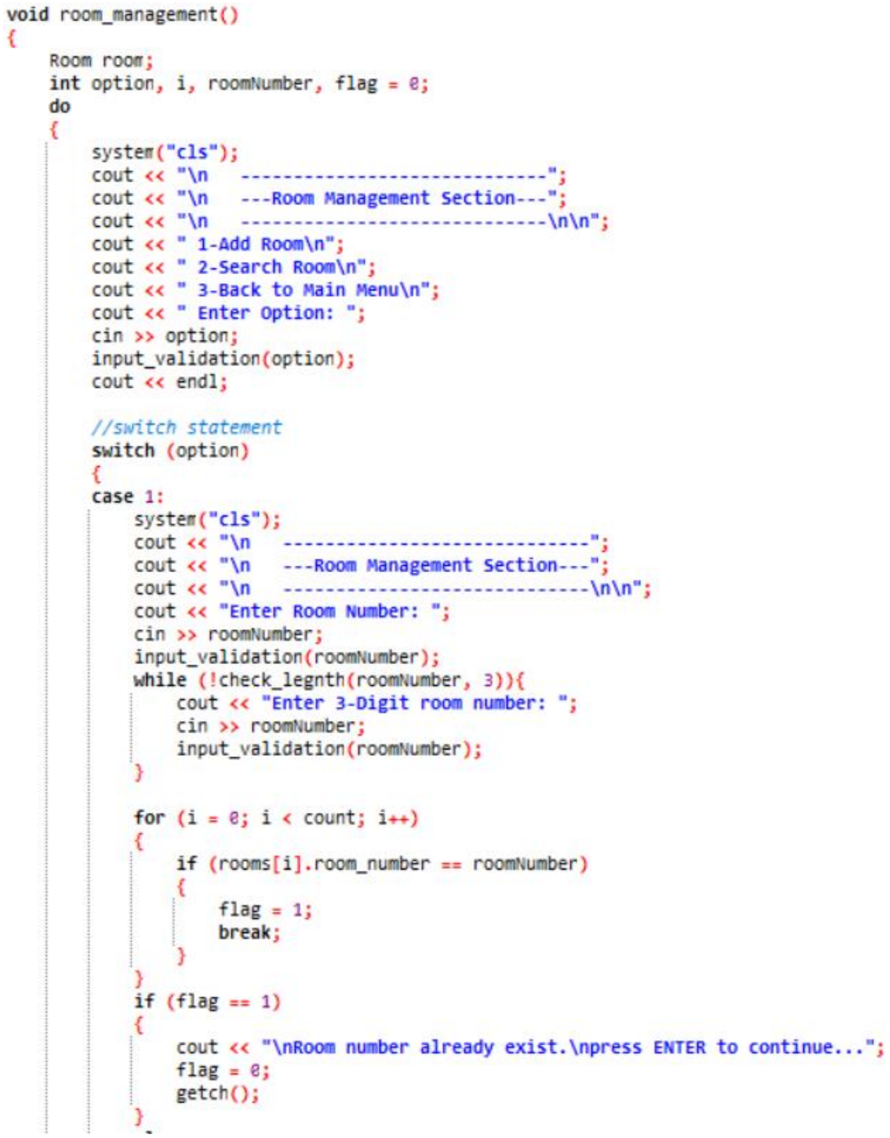
```
    void set_values(int booking_id, string name, int age, char gender, long long int phone_number, string city, int total_days, int advance_payment);


    int get_bookingID() {
        return booking_id;
    }
    string get_name() {
        return name;
    }
    string get_city() {
        return city;
    }
    int get_age() {
        return age;
    }
    char get_gender() {
        return gender;
    }
    long long int get_number() {
        return phone_number;
    }

    int get_days() {
        return total_days;
    }

    int get_advancePayment() {
        return advance_payment;
    }
};
```



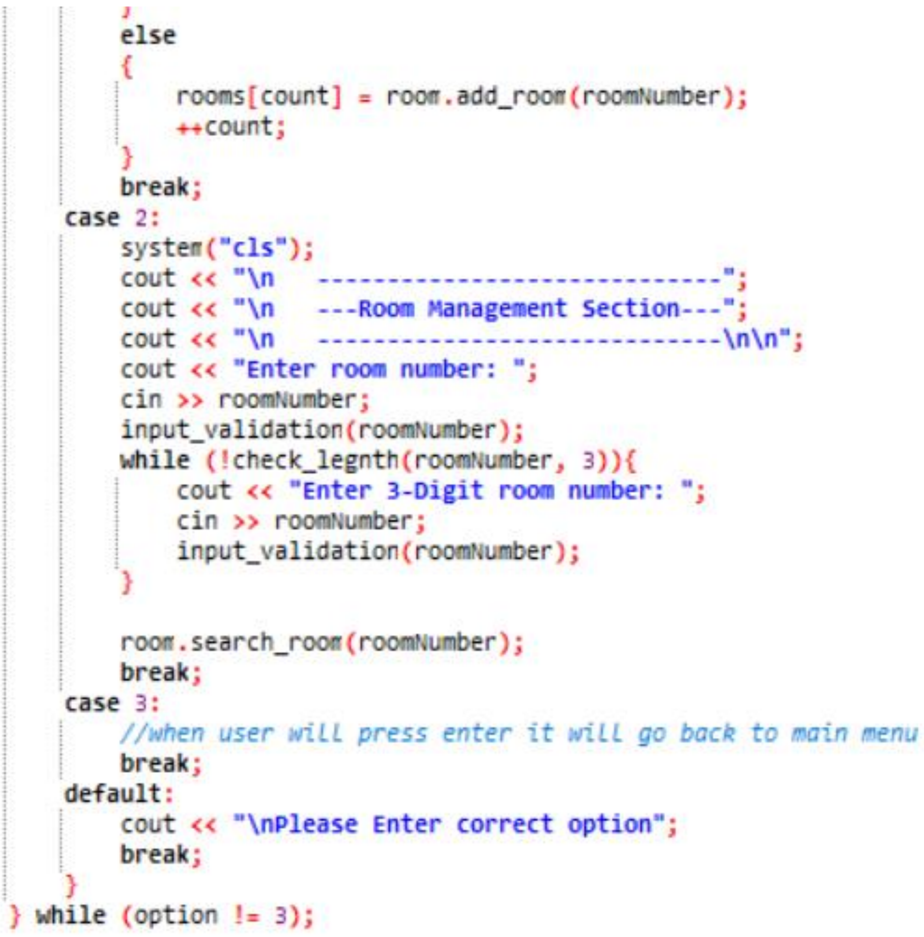
```
void customer_details::set_values(int booking_id, string name, int age, char gender, long long int phone_number, string city, int total_days, int advance_payment) {
    this->booking_id = booking_id;
    this->name = name;
    this->city = city;
    this->age = age;
    this->gender = gender;
    this->phone_number = phone_number;
    this->total_days = total_days;
    this->advance_payment = advance_payment;
}
```

## Room Management:

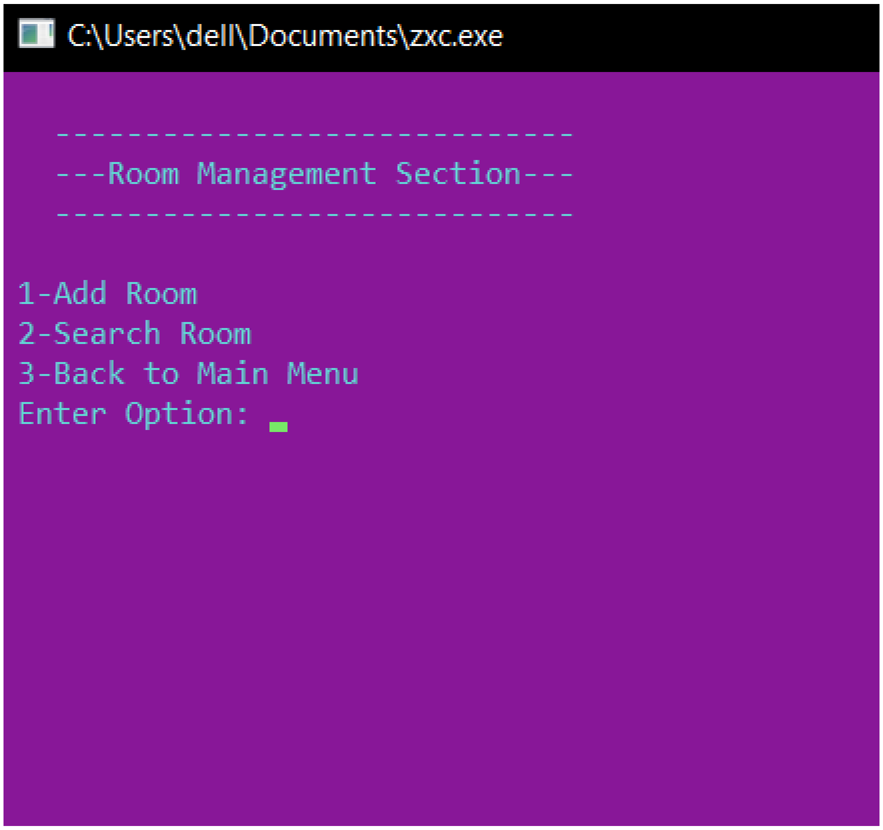
The room management function will give the available options for managing rooms such like add and search room it also has an option to go back to main menu .If the user chooses to add room it will ask for 3-digit room number which can not be repeated and take us to a function to add room details and if we select search room it will take us to the room searching function



```
void room_management()
{
    Room room;
    int option, i, roomNumber, flag = 0;
    do
    {
        system("cls");
        cout << "\n   ----------------------------";
        cout << "\n   ---Room Management Section---";
        cout << "\n   ----------------------------\n\n";
        cout << " 1-Add Room\n";
        cout << " 2-Search Room\n";
        cout << " 3-Back to Main Menu\n";
        cout << " Enter Option: ";
        cin >> option;
        input_validation(option);
        cout << endl;

        //switch statement
        switch (option)
        {
        case 1:
            system("cls");
            cout << "\n   ----------------------------";
            cout << "\n   ---Room Management Section---";
            cout << "\n   ----------------------------\n\n";
            cout << "Enter Room Number: ";
            cin >> roomNumber;
            input_validation(roomNumber);
            while (!check_legnth(roomNumber, 3)){
                cout << "Enter 3-Digit room number: ";
                cin >> roomNumber;
                input_validation(roomNumber);
            }

            for (i = 0; i < count; i++)
            {
                if (rooms[i].room_number == roomNumber)
                {
                    flag = 1;
                    break;
                }
            }
            if (flag == 1)
            {
                cout << "\nRoom number already exist.\npress ENTER to continue...";
                flag = 0;
                getch();
            }
```



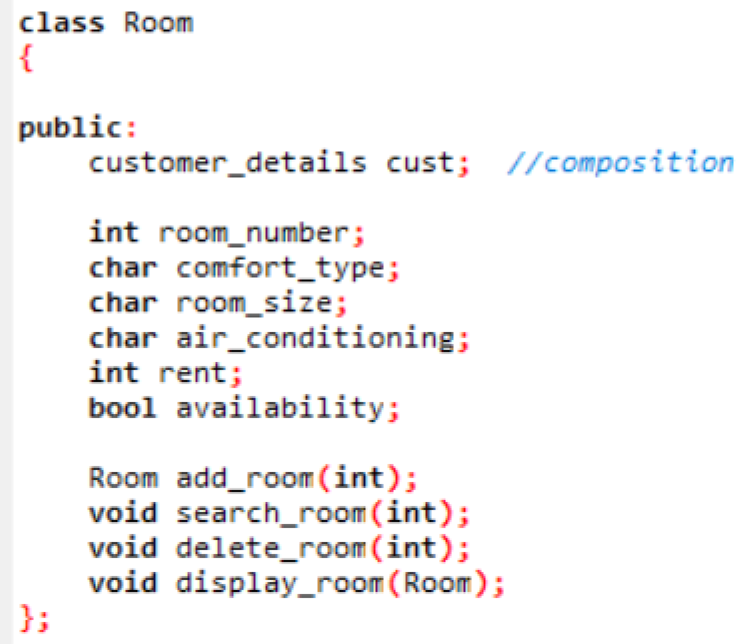
```
        }
        else
        {
            rooms[count] = room.add_room(roomNumber);
            ++count;
        }
        break;
    case 2:
        system("cls");
        cout << "\n   -----------------------------";
        cout << "\n   ---Room Management Section---";
        cout << "\n   -----------------------------\n\n";
        cout << "Enter room number: ";
        cin >> roomNumber;
        input_validation(roomNumber);
        while (!check_legnth(roomNumber, 3)){
            cout << "Enter 3-Digit room number: ";
            cin >> roomNumber;
            input_validation(roomNumber);
        }

        room.search_room(roomNumber);
        break;
    case 3:
        //when user will press enter it will go back to main menu
        break;
    default:
        cout << "\nPlease Enter correct option";
        break;
    }
} while (option != 3);
```



```
C:\Users\dell\Documents\zxc.exe

   -----------------------------
   ---Room Management Section---
   -----------------------------

1-Add Room
2-Search Room
3-Back to Main Menu
Enter Option: _
```

## Class Room:

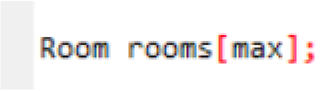
In the class customer we have done composition and created an object customer details cust and declared variables which take the information about rooms we have also declared 4 functions which are used to add ,search, delete and display room details



```
class Room
{

public:
    customer_details cust;  //composition

    int room_number;
    char comfort_type;
    char room_size;
    char air_conditioning;
    int rent;
    bool availability;

    Room add_room(int);
    void search_room(int);
    void delete_room(int);
    void display_room(Room);
};
```

## Array of objects:

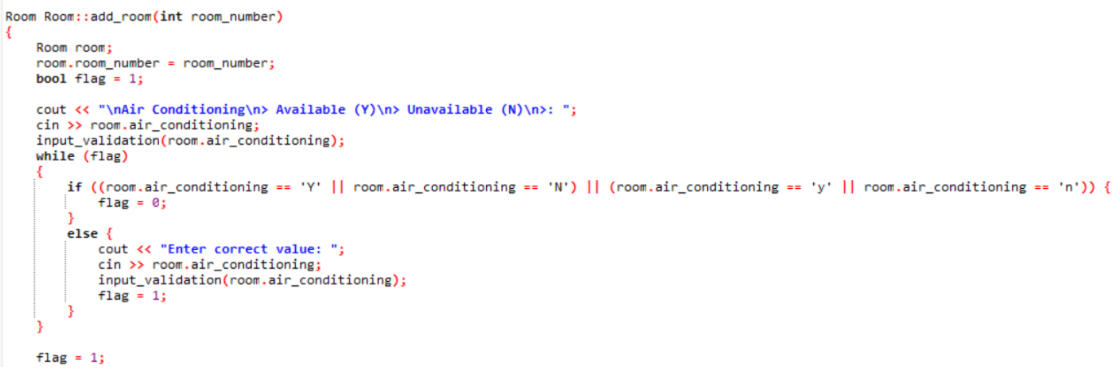
We have created an array of objects called room rooms



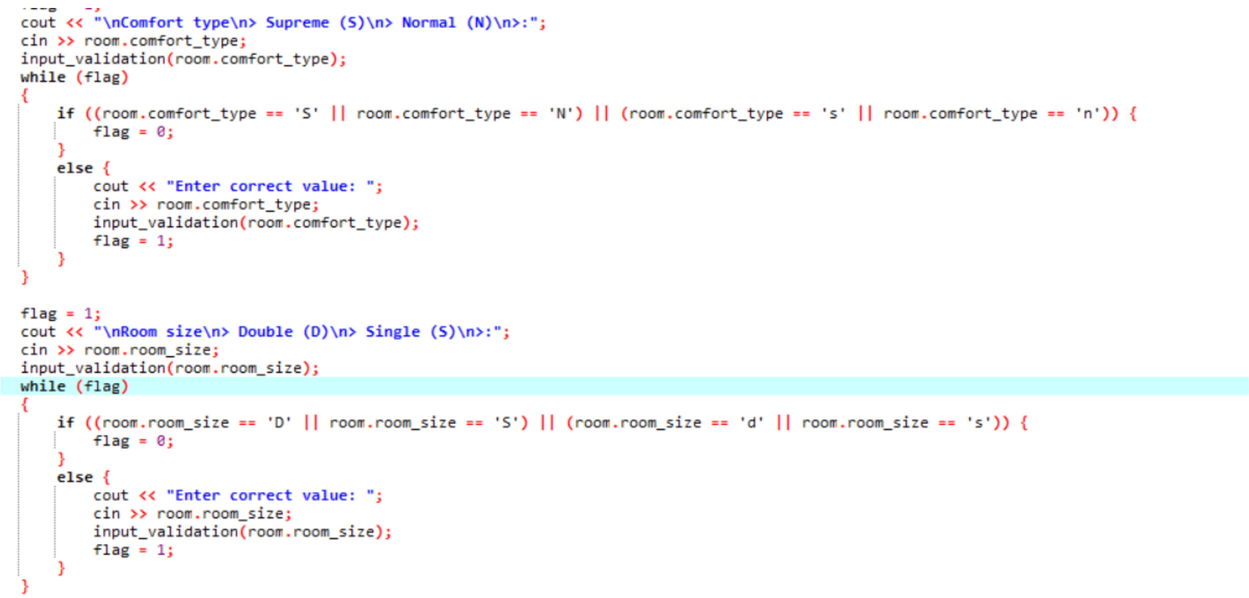
```
Room rooms[max];
```

## Room Room:

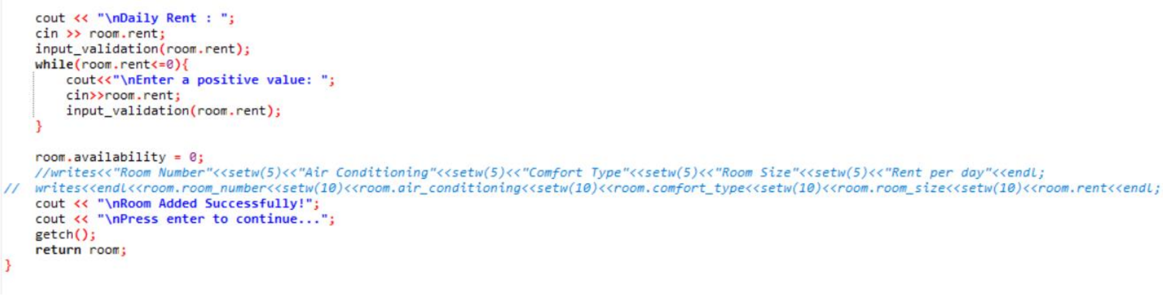
We have created a function room with return type room to add room in the system we have also created a temporary object room room In this function we will collect details about the room like room number, air condition,size,comfort type we are also using input validation by if else and bool flag so that only one the given options is selected



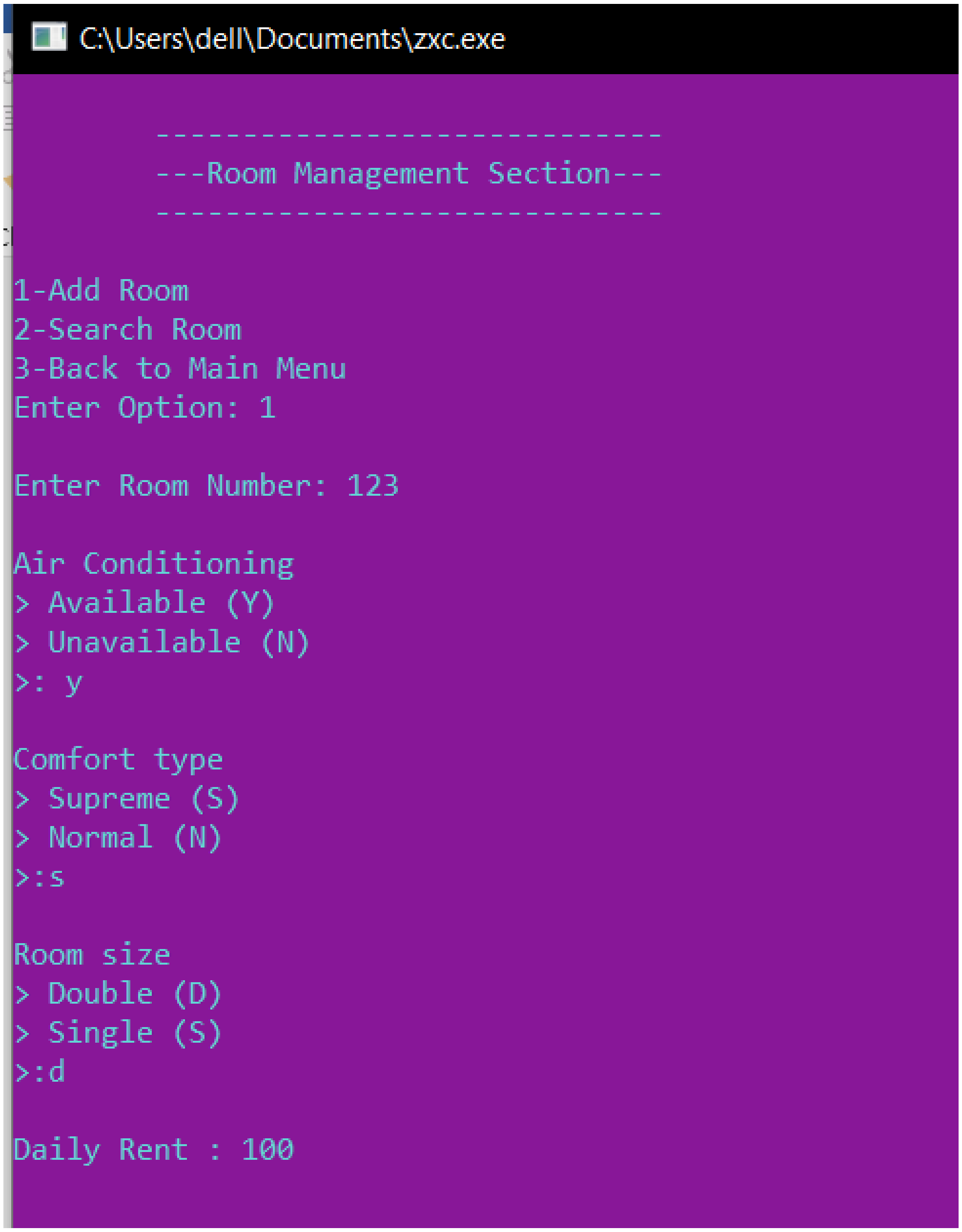
```
Room Room::add_room(int room_number)
{
    Room room;
    room.room_number = room_number;
    bool flag = 1;

    cout << "\nAir Conditioning\n> Available (Y)\n> Unavailable (N)\n>: ";
    cin >> room.air_conditioning;
    input_validation(room.air_conditioning);
    while (flag)
    {
        if ((room.air_conditioning == 'Y' || room.air_conditioning == 'N') || (room.air_conditioning == 'y' || room.air_conditioning == 'n')) {
            flag = 0;
        }
        else {
            cout << "Enter correct value: ";
            cin >> room.air_conditioning;
            input_validation(room.air_conditioning);
            flag = 1;
        }
    }

    flag = 1;
```



```
cout << "\nComfort type\n> Supreme (S)\n> Normal (N)\n>:";
cin >> room.comfort_type;
input_validation(room.comfort_type);
while (flag)
{
    if ((room.comfort_type == 'S' || room.comfort_type == 'N') || (room.comfort_type == 's' || room.comfort_type == 'n')) {
        flag = 0;
    }
    else {
        cout << "Enter correct value: ";
        cin >> room.comfort_type;
        input_validation(room.comfort_type);
        flag = 1;
    }
}

flag = 1;
cout << "\nRoom size\n> Double (D)\n> Single (S)\n>:";
cin >> room.room_size;
input_validation(room.room_size);
while (flag)
{
    if ((room.room_size == 'D' || room.room_size == 'S') || (room.room_size == 'd' || room.room_size == 's')) {
        flag = 0;
    }
    else {
        cout << "Enter correct value: ";
        cin >> room.room_size;
        input_validation(room.room_size);
        flag = 1;
    }
}
```



```
    cout << "\nDaily Rent : ";
    cin >> room.rent;
    input_validation(room.rent);
    while(room.rent<=0){
        cout<<"\nEnter a positive value: ";
        cin>>room.rent;
        input_validation(room.rent);
    }

    room.availability = 0;
    //writes<<"Room Number"<<setw(5)<<"Air Conditioning"<<setw(5)<<"Comfort Type"<<setw(5)<<"Room Size"<<setw(5)<<"Rent per day"<<endl;
//  writes<<endl<<room.room_number<<setw(10)<<room.air_conditioning<<setw(10)<<room.comfort_type<<setw(10)<<room.room_size<<setw(10)<<room.rent<<endl;
    cout << "\nRoom Added Successfully!";
    cout << "\nPress enter to continue...";
    getch();
    return room;
}
```



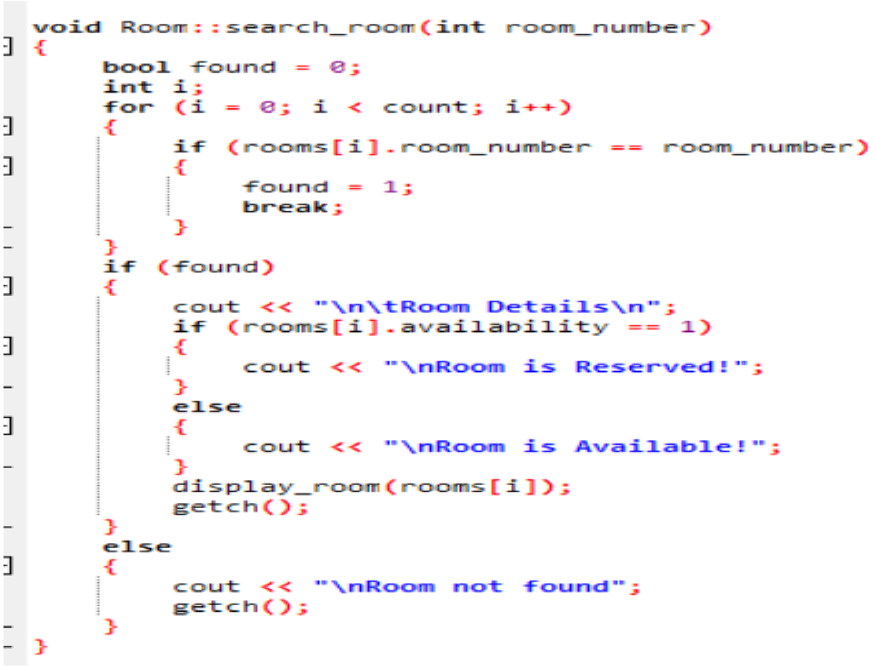
C:\Users\dell\Documents\zxc.exe

```
        -----------------------------
        ---Room Management Section---
        -----------------------------

1-Add Room
2-Search Room
3-Back to Main Menu
Enter Option: 1

Enter Room Number: 123

Air Conditioning
> Available (Y)
> Unavailable (N)
>: y

Comfort type
> Supreme (S)
> Normal (N)
>:s

Room size
> Double (D)
> Single (S)
>:d

Daily Rent : 100
```

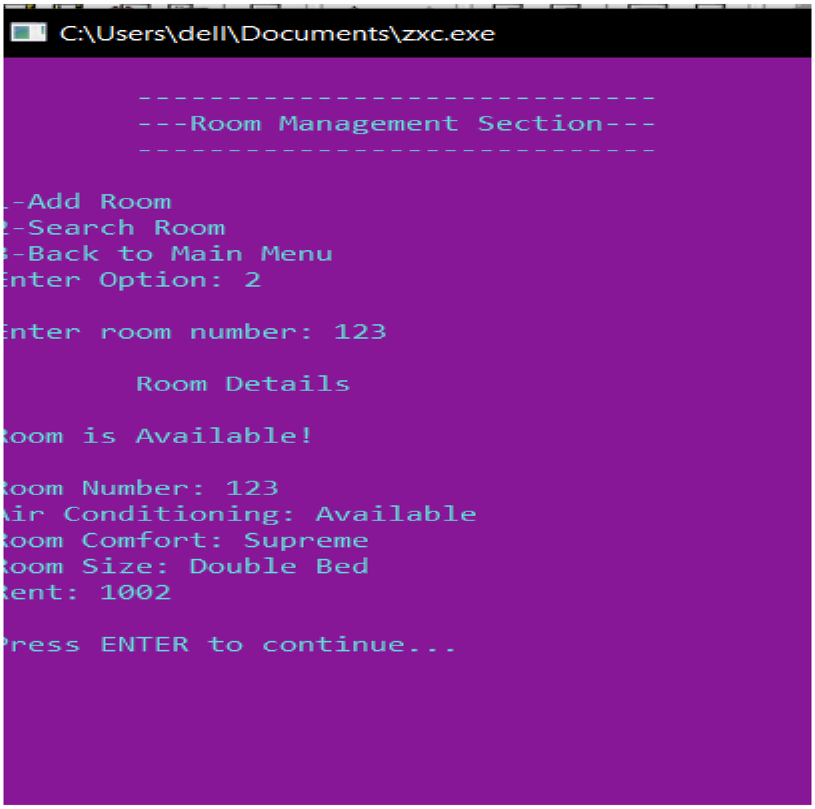
## Search Room, Display room:

The search room function is used to search the details of entered room using its room number if the room is not stored in data, it will show room not found and if it is stored it will check whether the room is available or reserved .It search for the room using for loop to search every index of array rooms

And then display the details using function display_room function



```
void Room::search_room(int room_number)
{
    bool found = 0;
    int i;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].room_number == room_number)
        {
            found = 1;
            break;
        }
    }
    if (found)
    {
        cout << "\n\tRoom Details\n";
        if (rooms[i].availability == 1)
        {
            cout << "\nRoom is Reserved!";
        }
        else
        {
            cout << "\nRoom is Available!";
        }
        display_room(rooms[i]);
        getch();
    }
    else
    {
        cout << "\nRoom not found";
        getch();
    }
}
```

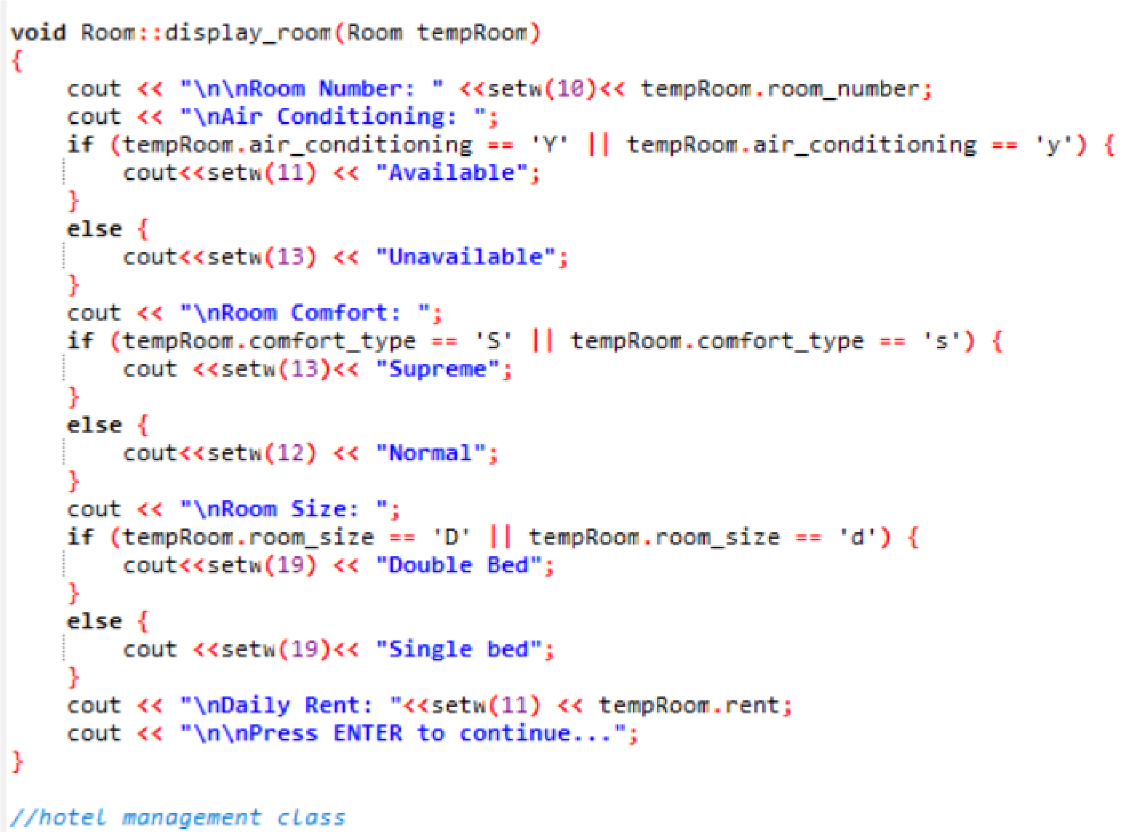


```
              -----------------------------
              ---Room Management Section---
              -----------------------------

-Add Room
-Search Room
-Back to Main Menu
nter Option: 2

nter room number: 123

              Room Details

oom is Available!

oom Number: 123
ir Conditioning: Available
oom Comfort: Supreme
oom Size: Double Bed
ent: 1002

ress ENTER to continue...
```

## Display room:

The display room function simply displays the details of the room number entered as shown above

We are also using the setw() function which is a member of iomanip library to set the field width of the output



```
void Room::display_room(Room tempRoom)
{
    cout << "\n\nRoom Number: " <<setw(10)<< tempRoom.room_number;
    cout << "\nAir Conditioning: ";
    if (tempRoom.air_conditioning == 'Y' || tempRoom.air_conditioning == 'y') {
        cout<<setw(11) << "Available";
    }
    else {
        cout<<setw(13) << "Unavailable";
    }
    cout << "\nRoom Comfort: ";
    if (tempRoom.comfort_type == 'S' || tempRoom.comfort_type == 's') {
        cout <<setw(13)<< "Supreme";
    }
    else {
        cout<<setw(12) << "Normal";
    }
    cout << "\nRoom Size: ";
    if (tempRoom.room_size == 'D' || tempRoom.room_size == 'd') {
        cout<<setw(19) << "Double Bed";
    }
    else {
        cout <<setw(19)<< "Single bed";
    }
    cout << "\nDaily Rent: "<<setw(11) << tempRoom.rent;
    cout << "\n\nPress ENTER to continue...";
}

//hotel management class
```

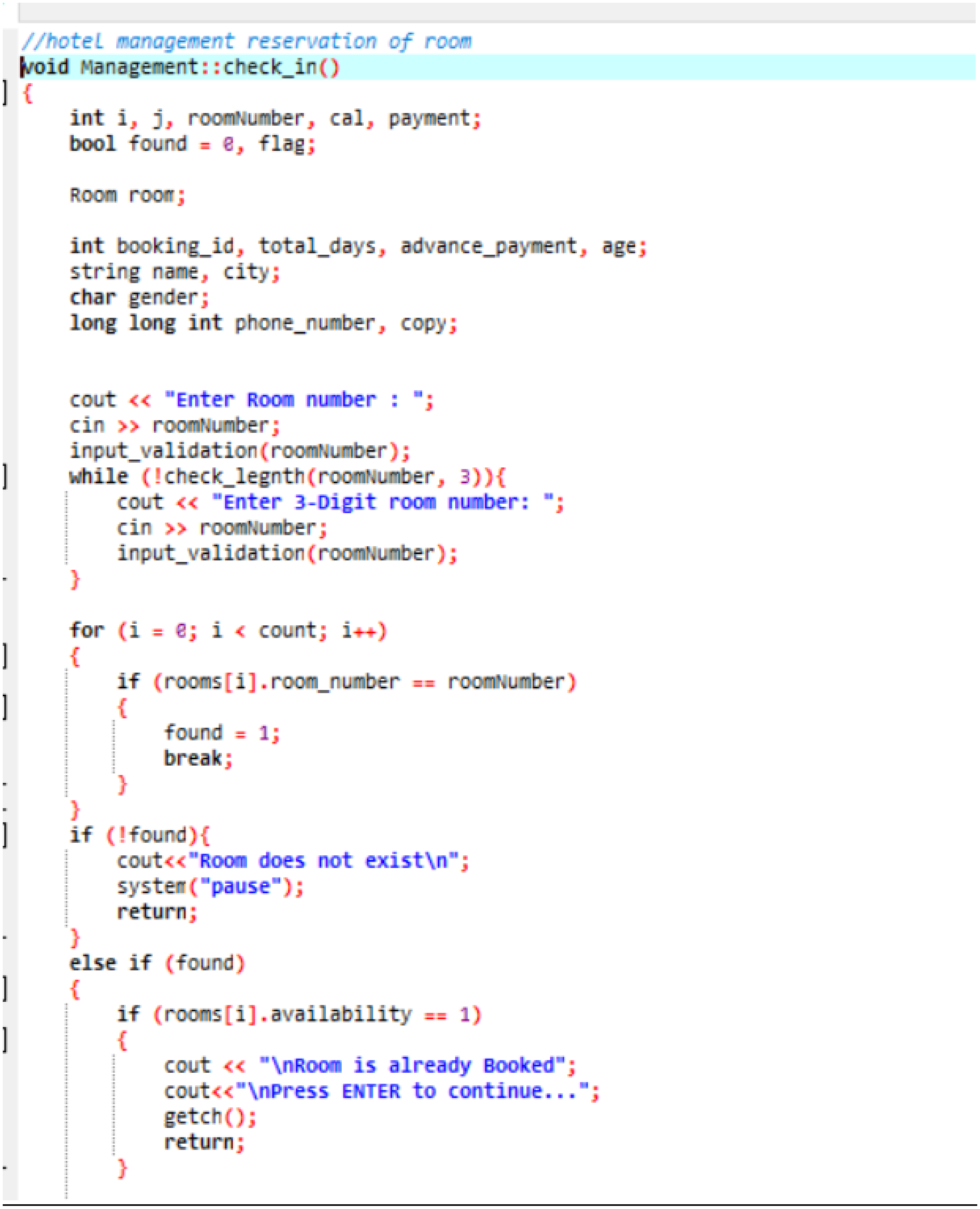
## Class Management:

The management class contains functions for void check_in,availabe_rooms,search_customer, check_out(int),customers_summary() and inherits members from room class

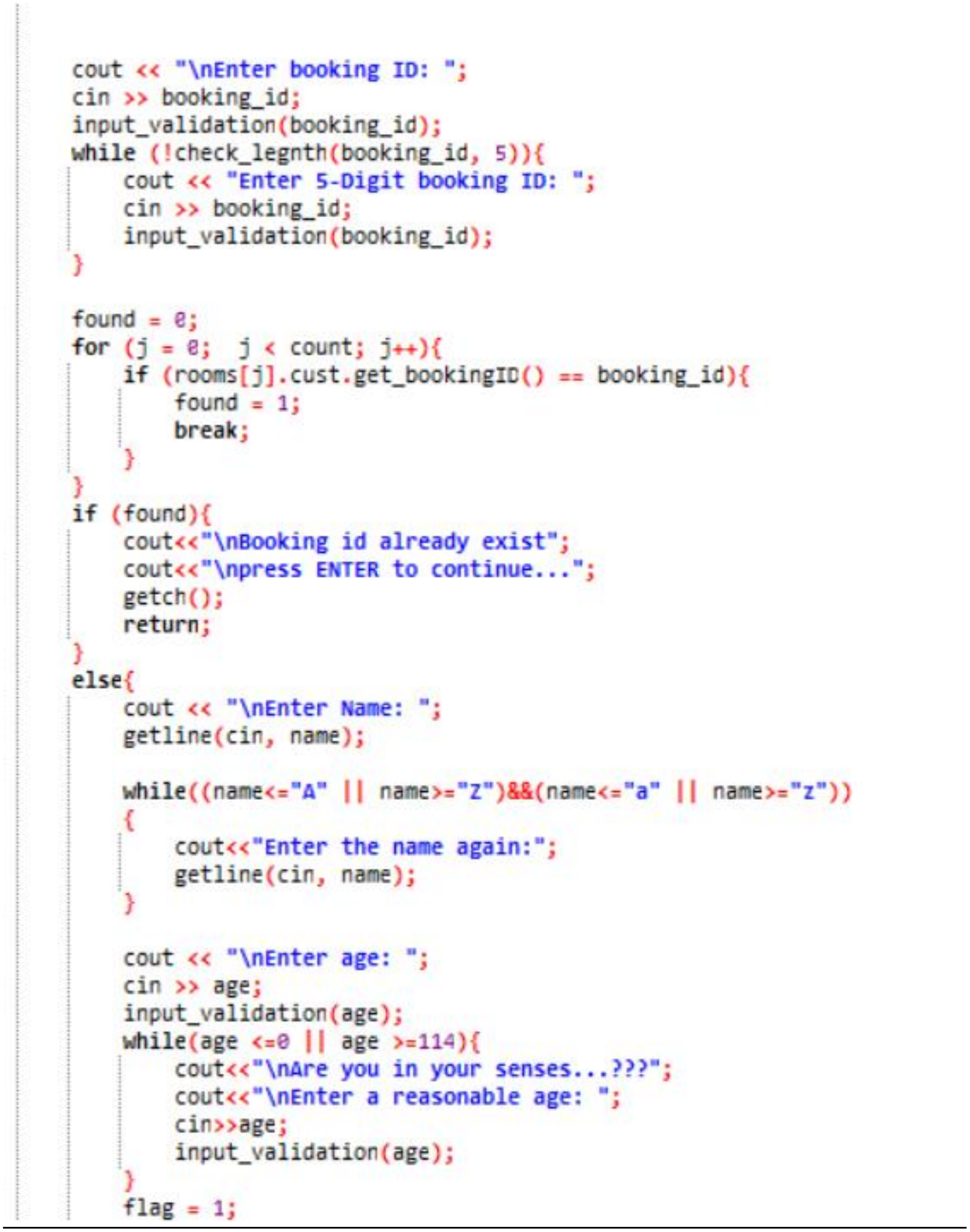
### Check IN:

The check in function will take the customers personal details like name, gender etc. and it will also ask for his age and check that the user does not input a too large value(max 114) or negative age (min 0).In this function we will also accept the advance payment the system will also keep in check that the advance payment in not greater than total bill. The maximum no of days a room can be booked is 99.

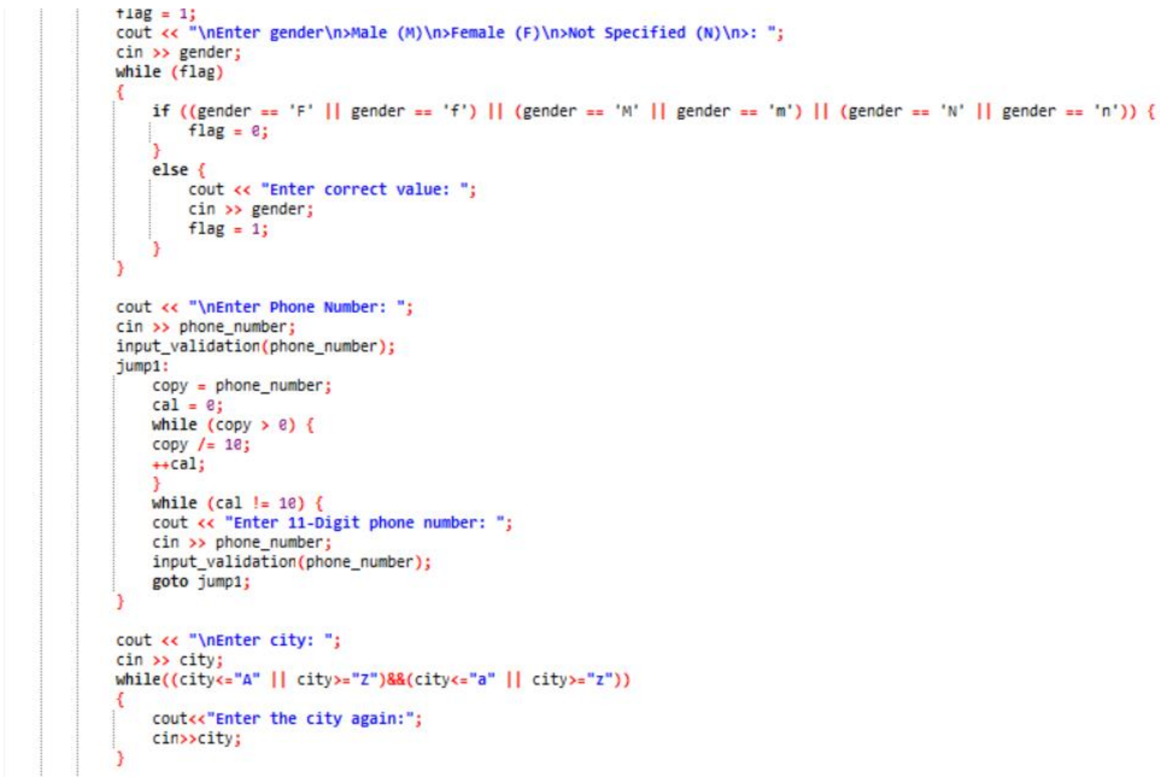
The system will also validate that a 11-digit phone number is entered .In this function we have used jump and goto statements to manipulate the flow of program if the conditions are not fulfilled



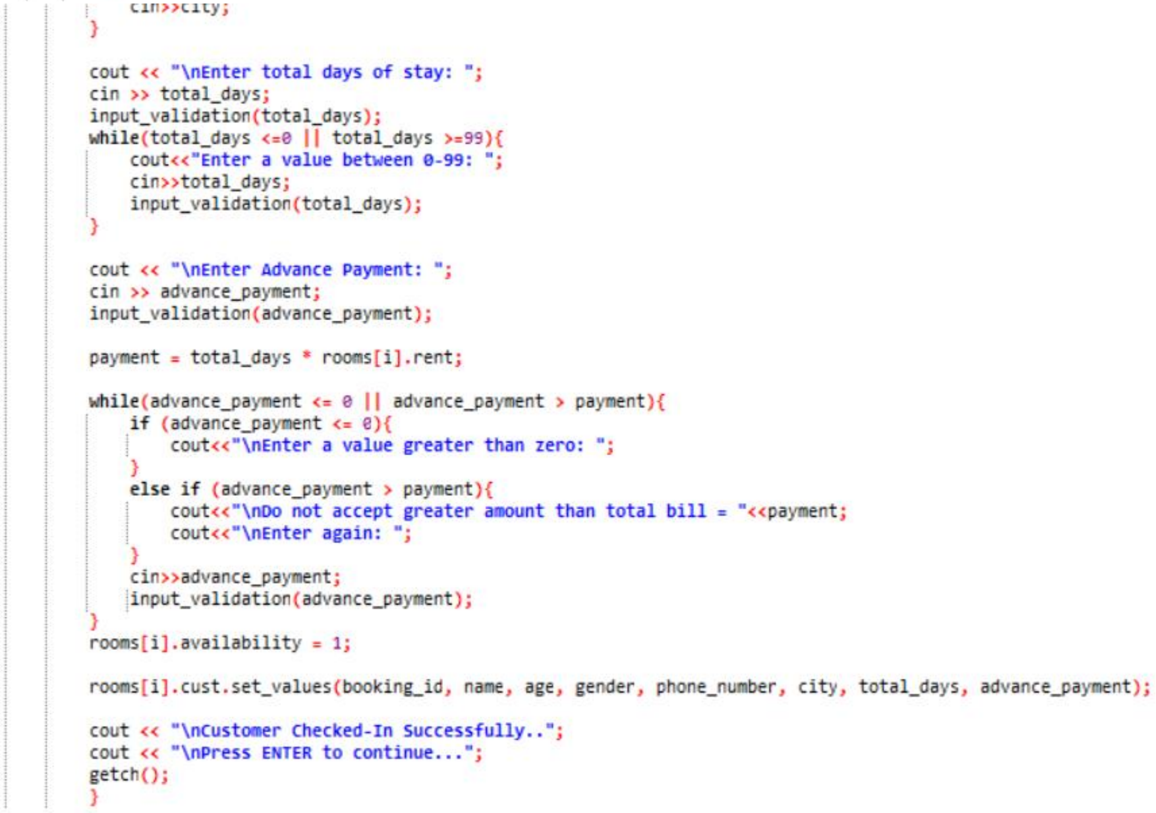
```
//hotel management reservation of room
void Management::check_in()
{
    int i, j, roomNumber, cal, payment;
    bool found = 0, flag;

    Room room;

    int booking_id, total_days, advance_payment, age;
    string name, city;
    char gender;
    long long int phone_number, copy;


    cout << "Enter Room number : ";
    cin >> roomNumber;
    input_validation(roomNumber);
    while (!check_legnth(roomNumber, 3)){
        cout << "Enter 3-Digit room number: ";
        cin >> roomNumber;
        input_validation(roomNumber);
    }

    for (i = 0; i < count; i++)
    {
        if (rooms[i].room_number == roomNumber)
        {
            found = 1;
            break;
        }
    }
    if (!found){
        cout<<"Room does not exist\n";
        system("pause");
        return;
    }
    else if (found)
    {
        if (rooms[i].availability == 1)
        {
            cout << "\nRoom is already Booked";
            cout<<"\nPress ENTER to continue...";
            getch();
            return;
        }
```



```
cout << "\nEnter booking ID: ";
cin >> booking_id;
input_validation(booking_id);
while (!check_legnth(booking_id, 5)){
    cout << "Enter 5-Digit booking ID: ";
    cin >> booking_id;
    input_validation(booking_id);
}

found = 0;
for (j = 0;  j < count; j++){
    if (rooms[j].cust.get_bookingID() == booking_id){
        found = 1;
        break;
    }
}
if (found){
    cout<<"\nBooking id already exist";
    cout<<"\npress ENTER to continue...";
    getch();
    return;
}
else{
    cout << "\nEnter Name: ";
    getline(cin, name);

    while((name<="A" || name>="Z")&&(name<="a" || name>="z"))
    {
        cout<<"Enter the name again:";
        getline(cin, name);
    }

    cout << "\nEnter age: ";
    cin >> age;
    input_validation(age);
    while(age <=0 || age >=114){
        cout<<"\nAre you in your senses...???";
        cout<<"\nEnter a reasonable age: ";
        cin>>age;
        input_validation(age);
    }
    flag = 1;
```

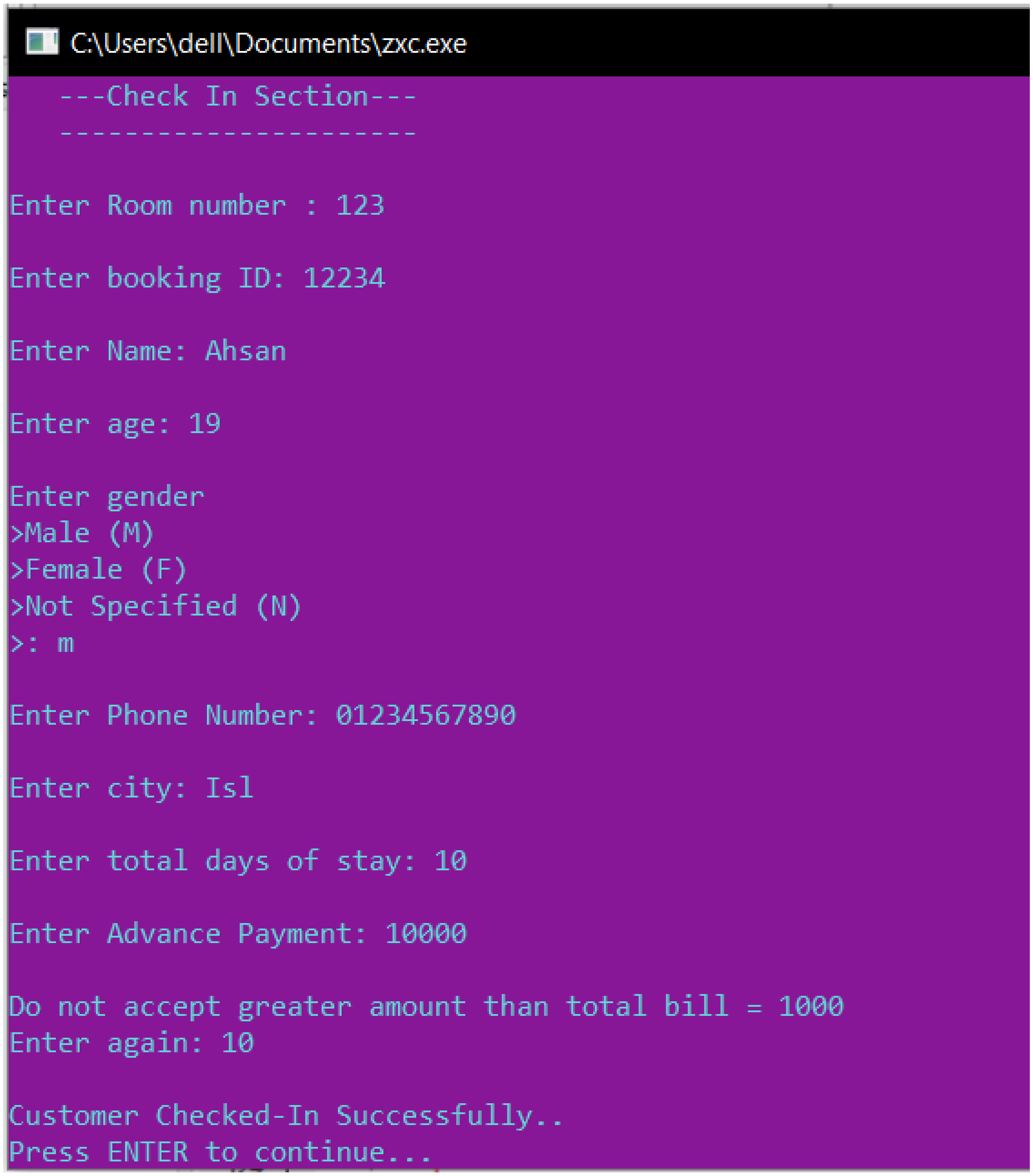


```cpp
flag = 1;
cout << "\nEnter gender\n>Male (M)\n>Female (F)\n>Not Specified (N)\n>: ";
cin >> gender;
while (flag)
{
    if ((gender == 'F' || gender == 'f') || (gender == 'M' || gender == 'm') || (gender == 'N' || gender == 'n')) {
        flag = 0;
    }
    else {
        cout << "Enter correct value: ";
        cin >> gender;
        flag = 1;
    }
}

cout << "\nEnter Phone Number: ";
cin >> phone_number;
input_validation(phone_number);
jump1:
    copy = phone_number;
    cal = 0;
    while (copy > 0) {
    copy /= 10;
    ++cal;
    }
    while (cal != 10) {
    cout << "Enter 11-Digit phone number: ";
    cin >> phone_number;
    input_validation(phone_number);
    goto jump1;
}

cout << "\nEnter city: ";
cin >> city;
while((city<="A" || city>="Z")&&(city<="a" || city>="z"))
{
    cout<<"Enter the city again:";
    cin>>city;
}
```



```cpp
    cin>>city;
}

cout << "\nEnter total days of stay: ";
cin >> total_days;
input_validation(total_days);
while(total_days <=0 || total_days >=99){
    cout<<"Enter a value between 0-99: ";
    cin>>total_days;
    input_validation(total_days);
}

cout << "\nEnter Advance Payment: ";
cin >> advance_payment;
input_validation(advance_payment);

payment = total_days * rooms[i].rent;

while(advance_payment <= 0 || advance_payment > payment){
    if (advance_payment <= 0){
        cout<<"\nEnter a value greater than zero: ";
    }
    else if (advance_payment > payment){
        cout<<"\nDo not accept greater amount than total bill = "<<payment;
        cout<<"\nEnter again: ";
    }
    cin>>advance_payment;
    input_validation(advance_payment);
}
rooms[i].availability = 1;

rooms[i].cust.set_values(booking_id, name, age, gender, phone_number, city, total_days, advance_payment);

cout << "\nCustomer Checked-In Successfully..";
cout << "\nPress ENTER to continue...";
getch();
}
```

```cpp
}
```

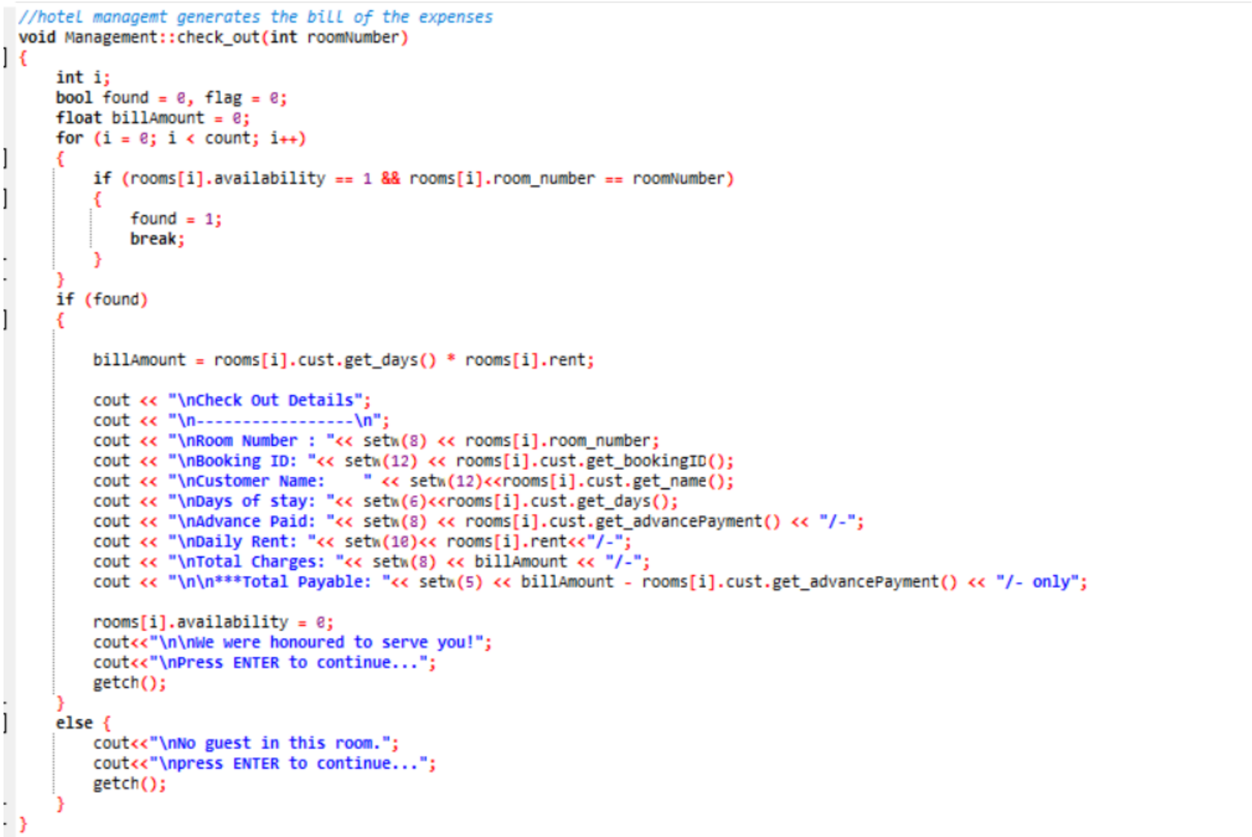


```
   ---Check In Section---
   ----------------------

Enter Room number : 123

Enter booking ID: 12234

Enter Name: Ahsan

Enter age: 19

Enter gender
>Male (M)
>Female (F)
>Not Specified (N)
>: m

Enter Phone Number: 01234567890

Enter city: Isl

Enter total days of stay: 10

Enter Advance Payment: 10000

Do not accept greater amount than total bill = 1000
Enter again: 10

Customer Checked-In Successfully..
Press ENTER to continue...
```

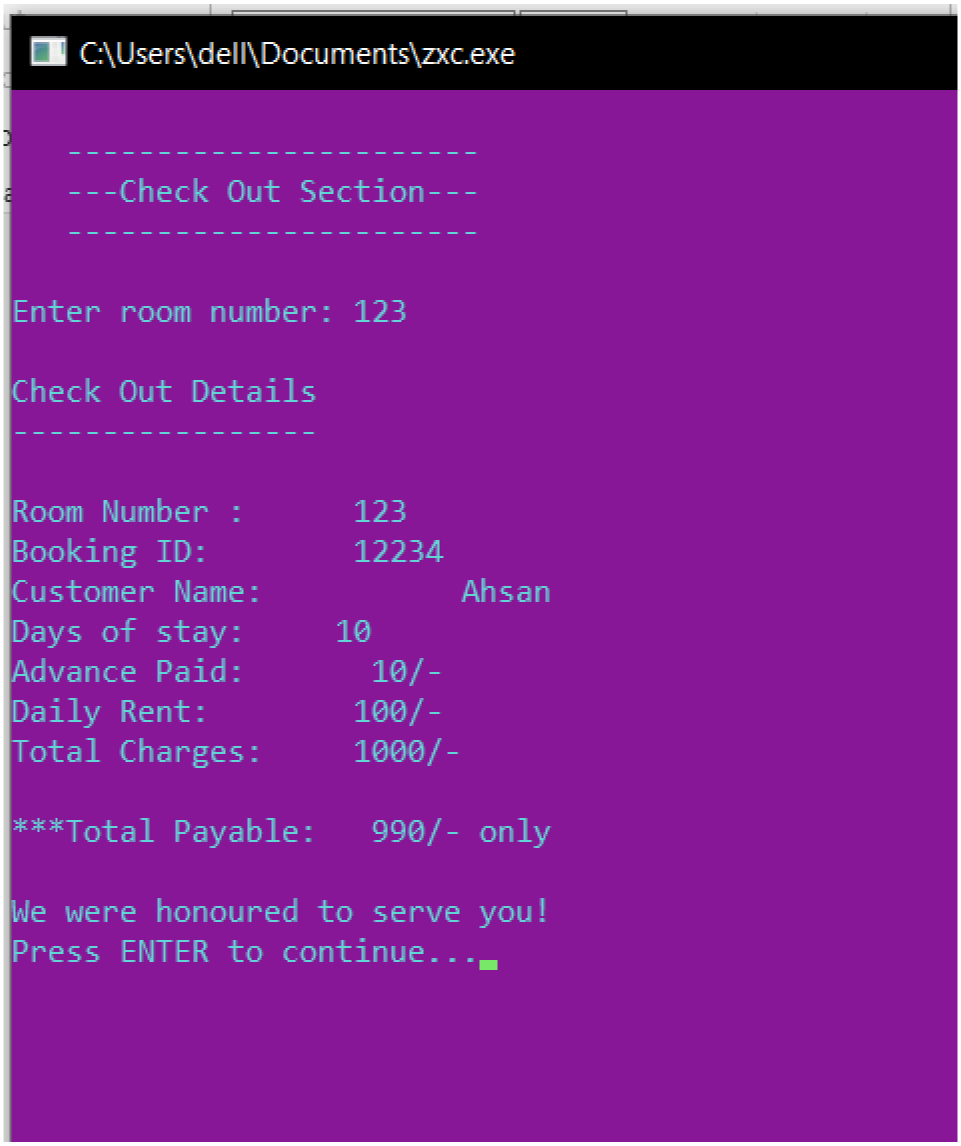
## Check Out:

The check out function will simply display the customer details and total bill after subtracting the rent from advance payment

After the person checks out the room will be once again available for customers



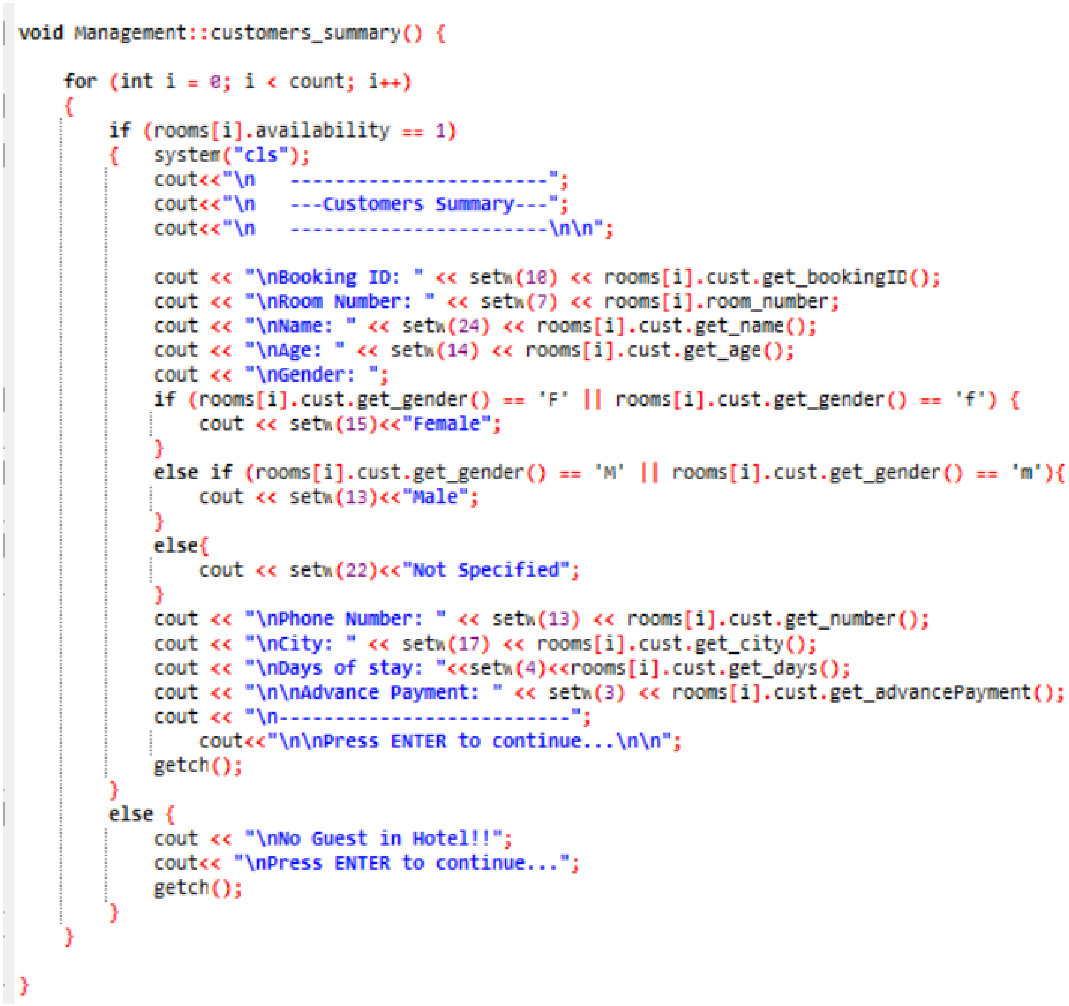
```cpp
//hotel managemt generates the bill of the expenses
void Management::check_out(int roomNumber)
{
    int i;
    bool found = 0, flag = 0;
    float billAmount = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].availability == 1 && rooms[i].room_number == roomNumber)
        {
            found = 1;
            break;
        }
    }
    if (found)
    {

        billAmount = rooms[i].cust.get_days() * rooms[i].rent;

        cout << "\nCheck Out Details";
        cout << "\n-----------------\n";
        cout << "\nRoom Number : "<< setw(8) << rooms[i].room_number;
        cout << "\nBooking ID: "<< setw(12) << rooms[i].cust.get_bookingID();
        cout << "\nCustomer Name:    " << setw(12)<<rooms[i].cust.get_name();
        cout << "\nDays of stay: "<< setw(6)<<rooms[i].cust.get_days();
        cout << "\nAdvance Paid: "<< setw(8) << rooms[i].cust.get_advancePayment() << "/-";
        cout << "\nDaily Rent: "<< setw(10)<< rooms[i].rent<<"/-";
        cout << "\nTotal Charges: "<< setw(8) << billAmount << "/-";
        cout << "\n\n***Total Payable: "<< setw(5) << billAmount - rooms[i].cust.get_advancePayment() << "/- only";

        rooms[i].availability = 0;
        cout<<"\n\nWe were honoured to serve you!";
        cout<<"\nPress ENTER to continue...";
        getch();
    }
    else {
        cout<<"\nNo guest in this room.";
        cout<<"\npress ENTER to continue...";
        getch();
    }
}
```



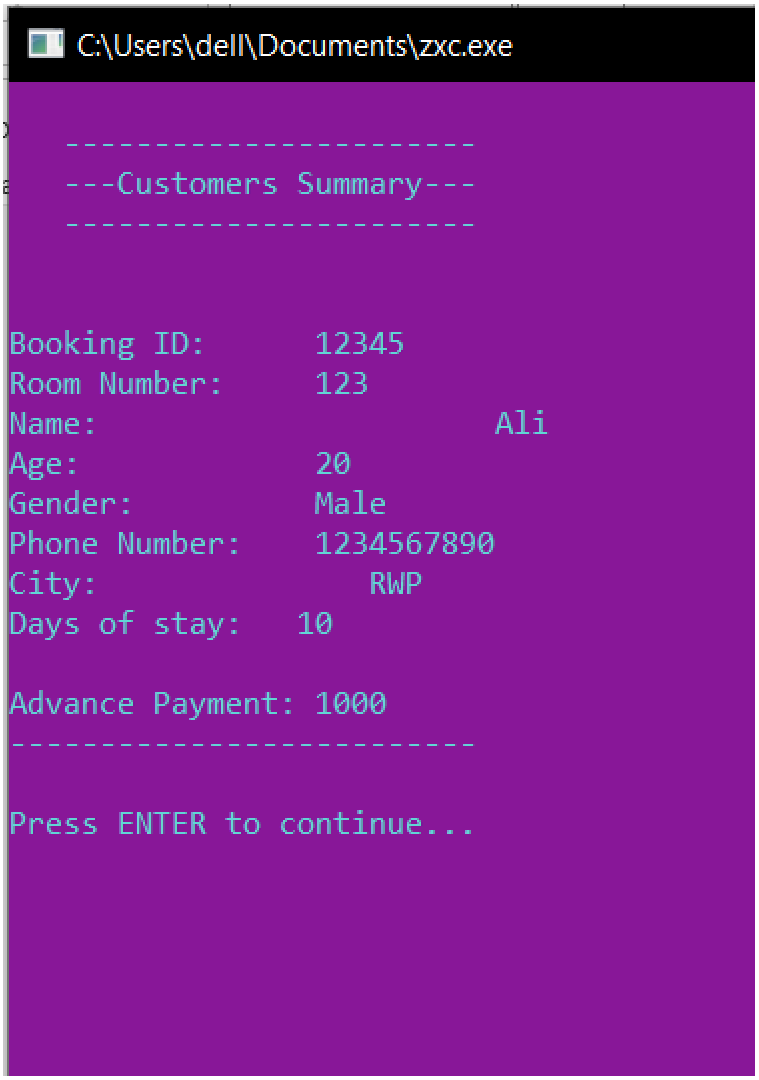
```
   -----------------------
   ---Check Out Section---
   -----------------------

Enter room number: 123

Check Out Details
-----------------

Room Number :      123
Booking ID:        12234
Customer Name:           Ahsan
Days of stay:     10
Advance Paid:       10/-
Daily Rent:        100/-
Total Charges:     1000/-

***Total Payable:   990/- only

We were honoured to serve you!
Press ENTER to continue...
```

## Customer Summary:

The customer summary function will check every index of the array rooms and display the details of every customer currently staying in the hotel once a customer checks out the data will automatically be removed



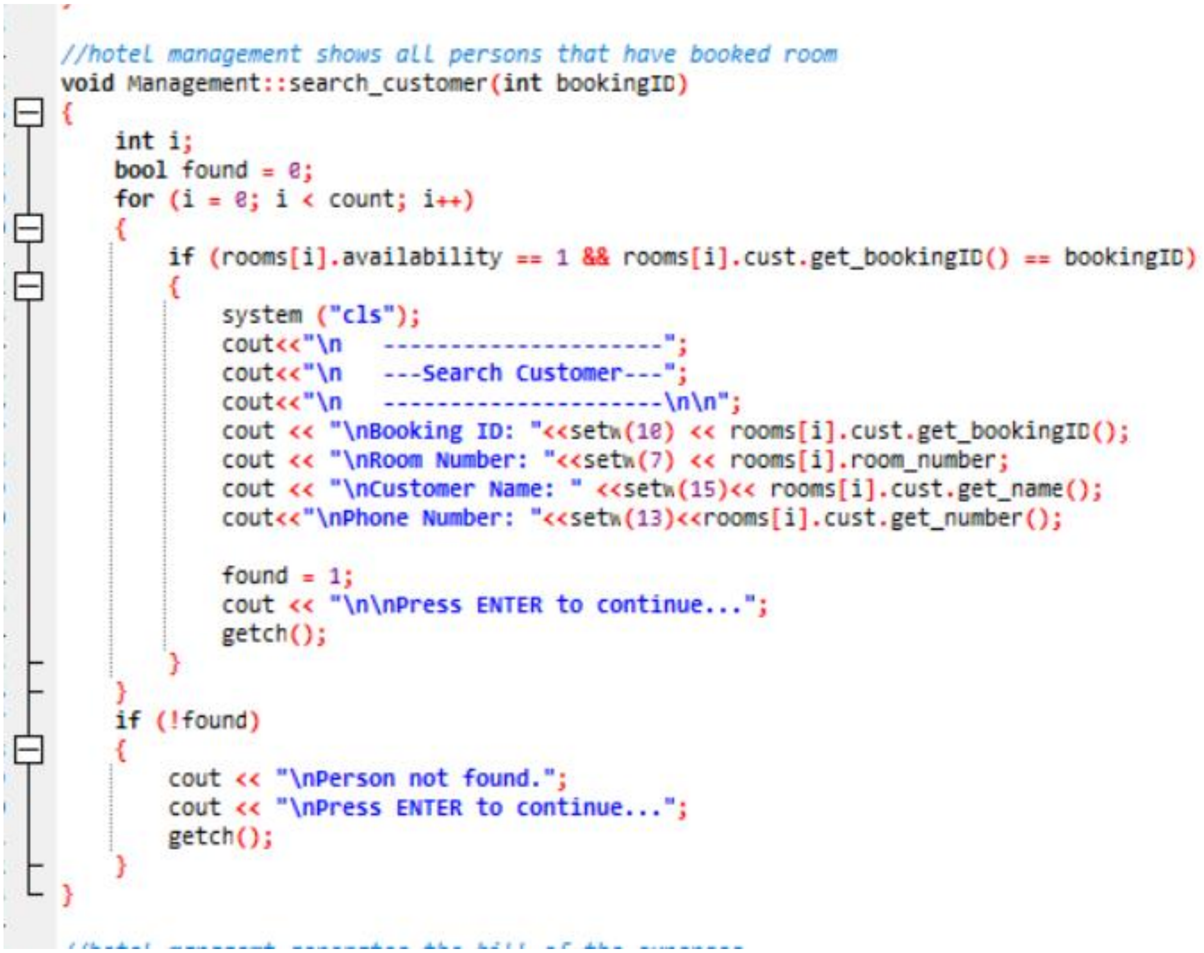
```
void Management::customers_summary() {

    for (int i = 0; i < count; i++)
    {
        if (rooms[i].availability == 1)
        {   system("cls");
            cout<<"\n   -----------------------";
            cout<<"\n   ---Customers Summary---";
            cout<<"\n   -----------------------\n\n";

            cout << "\nBooking ID: " << setw(10) << rooms[i].cust.get_bookingID();
            cout << "\nRoom Number: " << setw(7) << rooms[i].room_number;
            cout << "\nName: " << setw(24) << rooms[i].cust.get_name();
            cout << "\nAge: " << setw(14) << rooms[i].cust.get_age();
            cout << "\nGender: ";
            if (rooms[i].cust.get_gender() == 'F' || rooms[i].cust.get_gender() == 'f') {
                cout << setw(15)<<"Female";
            }
            else if (rooms[i].cust.get_gender() == 'M' || rooms[i].cust.get_gender() == 'm'){
                cout << setw(13)<<"Male";
            }
            else{
                cout << setw(22)<<"Not Specified";
            }
            cout << "\nPhone Number: " << setw(13) << rooms[i].cust.get_number();
            cout << "\nCity: " << setw(17) << rooms[i].cust.get_city();
            cout << "\nDays of stay: "<<setw(4)<<rooms[i].cust.get_days();
            cout << "\n\nAdvance Payment: " << setw(3) << rooms[i].cust.get_advancePayment();
            cout << "\n-------------------------";
                cout<<"\n\nPress ENTER to continue...\n\n";
            getch();
        }
        else {
            cout << "\nNo Guest in Hotel!!";
            cout<< "\nPress ENTER to continue...";
            getch();
        }
    }

}
```



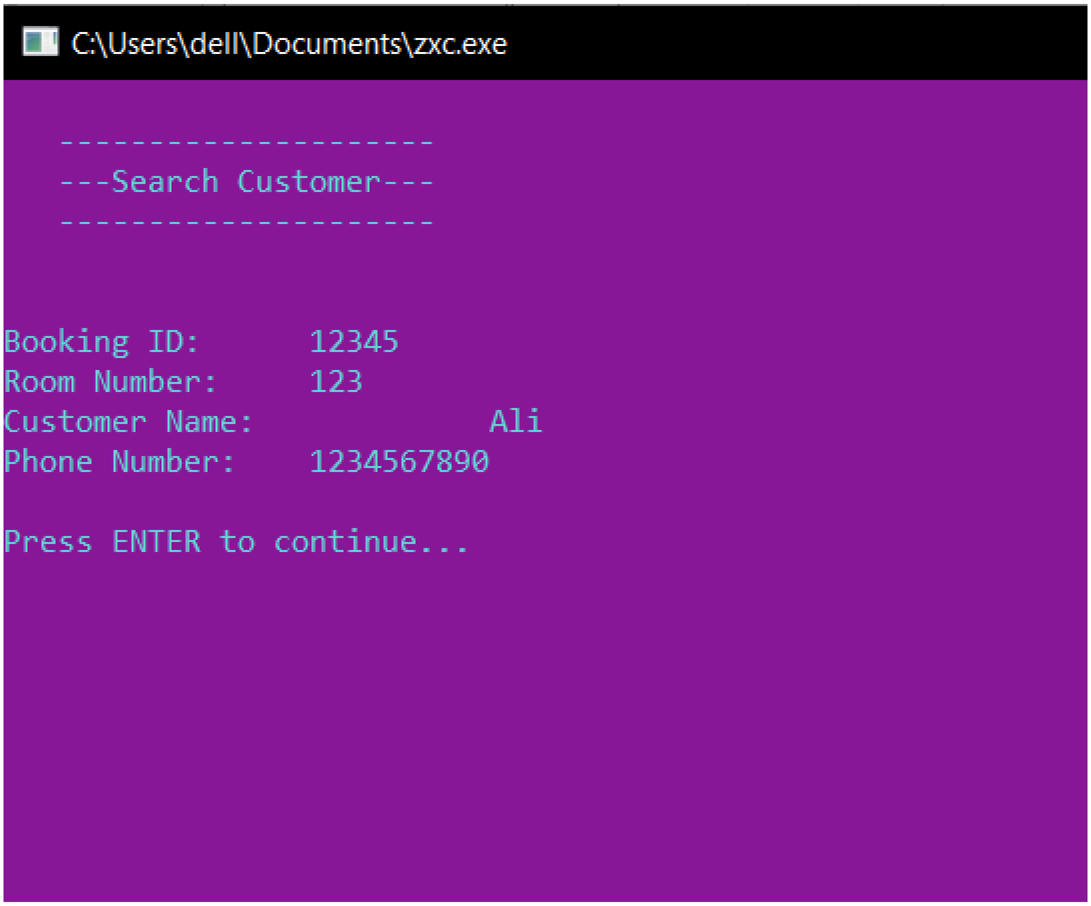
```
   -----------------------
   ---Customers Summary---
   -----------------------


Booking ID:      12345
Room Number:     123
Name:                      Ali
Age:             20
Gender:          Male
Phone Number:    1234567890
City:               RWP
Days of stay:   10

Advance Payment: 1000
--------------------------

Press ENTER to continue...
```

## Search Customer:

The search customer function will ask for the customer id and search for the id in the array of stored data .Once it finds a matching booking id it displays the details of the relevant customer



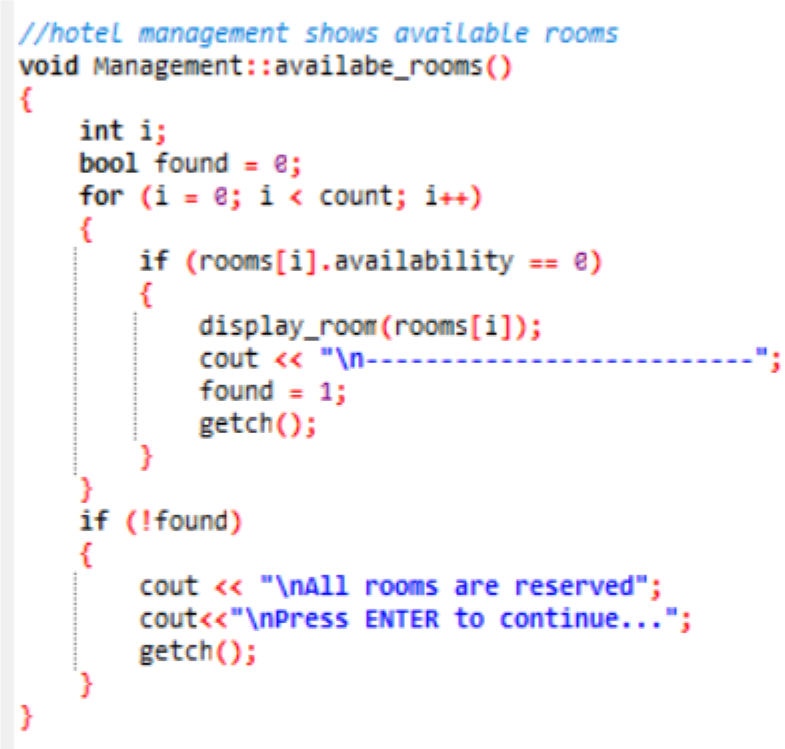
```
//hotel management shows all persons that have booked room
void Management::search_customer(int bookingID)
{
    int i;
    bool found = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].availability == 1 && rooms[i].cust.get_bookingID() == bookingID)
        {
            system ("cls");
            cout<<"\n   --------------------";
            cout<<"\n   ---Search Customer---";
            cout<<"\n   ---------------------\n\n";
            cout << "\nBooking ID: "<<setw(10) << rooms[i].cust.get_bookingID();
            cout << "\nRoom Number: "<<setw(7) << rooms[i].room_number;
            cout << "\nCustomer Name: " <<setw(15)<< rooms[i].cust.get_name();
            cout<<"\nPhone Number: "<<setw(13)<<rooms[i].cust.get_number();

            found = 1;
            cout << "\n\nPress ENTER to continue...";
            getch();
        }
    }
    if (!found)
    {
        cout << "\nPerson not found.";
        cout << "\nPress ENTER to continue...";
        getch();
    }
}

//hotel managment generates the bill of the expenses
```



```
   ---------------------
   ---Search Customer---
   ---------------------


Booking ID:      12345
Room Number:     123
Customer Name:             Ali
Phone Number:    1234567890

Press ENTER to continue...
```

## Available rooms:

The available room's function will display all the unoccupied rooms in the hotel .If no room is entered it will ask you to add room or if no room is available, it will tell us that all rooms are reserved.



```
//hotel management shows available rooms
void Management::availabe_rooms()
{
    int i;
    bool found = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].availability == 0)
        {
            display_room(rooms[i]);
            cout << "\n-------------------------";
            found = 1;
            getch();
        }
    }
    if (!found)
    {
        cout << "\nAll rooms are reserved";
        cout<<"\nPress ENTER to continue...";
        getch();
    }
}
```

## Templates

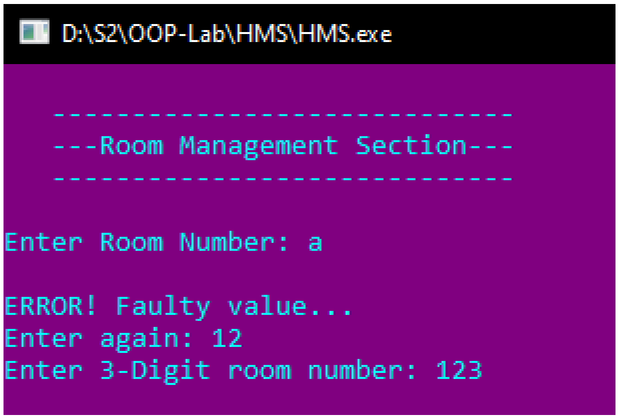
We have used two functions using templates. Firs one is input_validation() and other one is check_length().

First function is used to check If input is of right datatype or not. While other is used to calculate the length of integers to get the required length.

**Code**

```
template <typename T>
void input_validation(T& value){

    while(!cin.good()){
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout<<"\nERROR! Faulty value...";
        cout<<"\nEnter again: ";
        cin>>value;
    }
    cin.clear();
    cin.ignore(INT_MAX, '\n');
}

template <typename U>
bool check_legnth(U& num, U digit){
    U count, cal, copy;
jump:
    copy = num;
    cal = 0;
    while (copy > 0) {
        copy /= 10;
        ++cal;
    }
    while (cal != digit) {
        return false;
        goto jump;
    }
    return true;
}
```

**Console**



```
   -----------------------------
   ---Room Management Section---
   -----------------------------

Enter Room Number: a

ERROR! Faulty value...
Enter again: 12
Enter 3-Digit room number: 123
```