

**UMT**

**Object-Oriented Programming Semester Project Proposal**

**Submitted By:**

Asra Shakeel (71)

Mirha Hanan (42)

Atiqa Akbar (46)

Zarmeen Tauseef (52)

**Submitted to:**

Sir Talha

**Project Title: Hotel Management System**

**Introduction:**

The field of Object-Oriented Programming (OOP) is fundamental to modern software development, providing a modular and scalable approach to designing and organizing code. This semester project aims to explore key concepts of OOP through practical application, fostering a deeper

understanding of its principles.

**Objectives:**

To gain hands-on experience in designing and implementing object-oriented solutions.

To develop proficiency in C++ programming language.

To understand the importance of classes, inheritance, and constructor in software development.

**Project Title: Hotel Management System**

The Hotel Management System is a software application designed to facilitate the management of hotel bookings, room reservations, guest details, and payments. This project aims to provide a comprehensive solution to automate various hotel management tasks, thereby enhancing efficiency and improving the overall guest experience.

**Methodology:**

1. **Guest Management:**

Store guest details (name, surname, age)

Display guest information

1. **Room Management:**

Store room details (room ID, room type)

Display room information

1. **Booking Management:**

Store booking details (booking ID, date from, date to, number of guests)

Display booking information

1. **Payment Management:**

Store payment details (amount, card number)

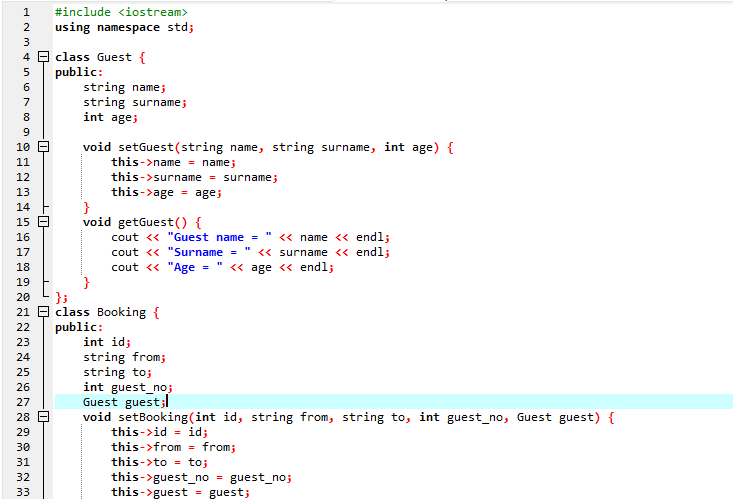
Display payment information

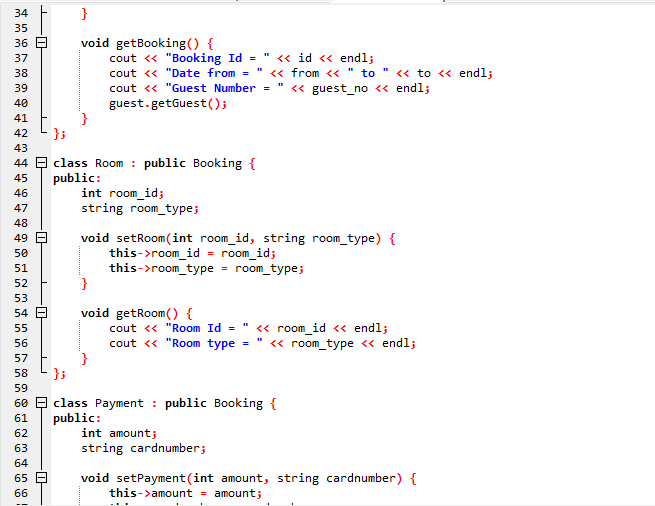
1. **Main function:**

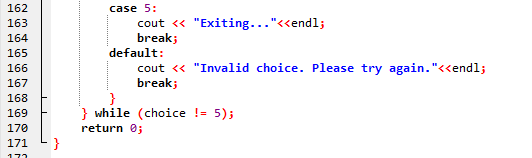
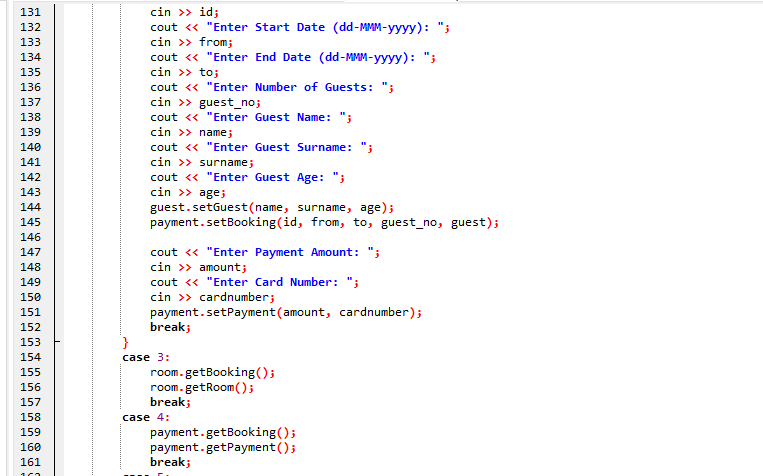
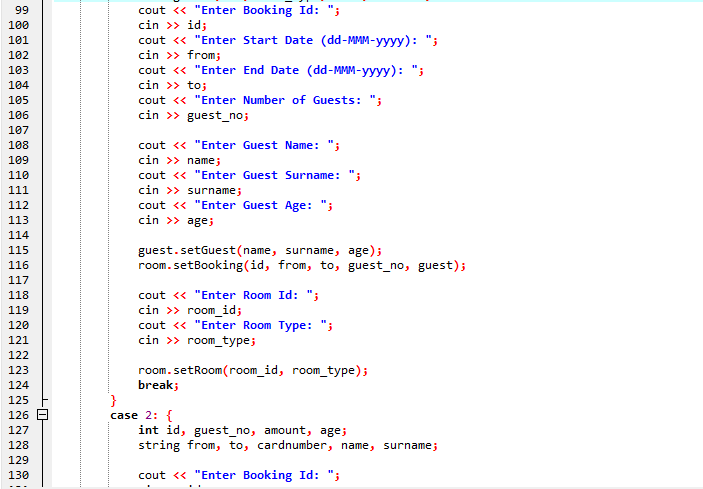
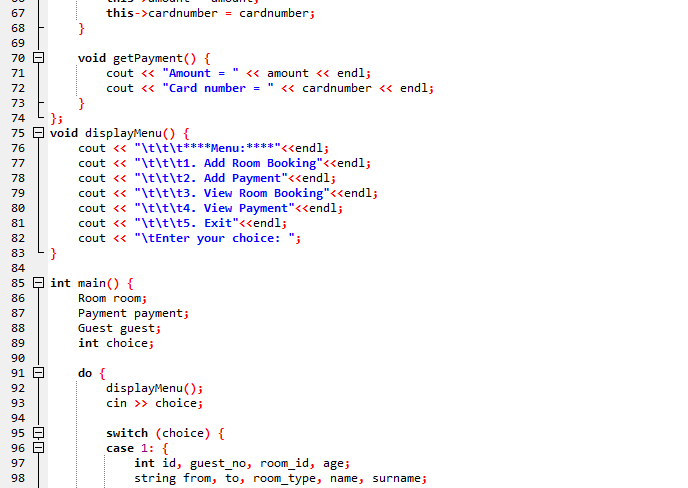
Creates instances of Room and Payment.

Calls the respective methods to display the details of each instance.

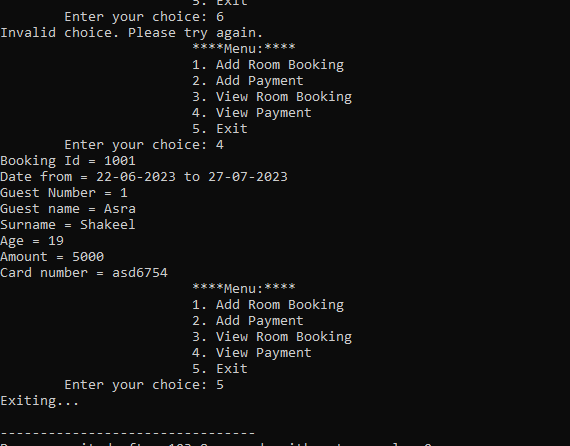
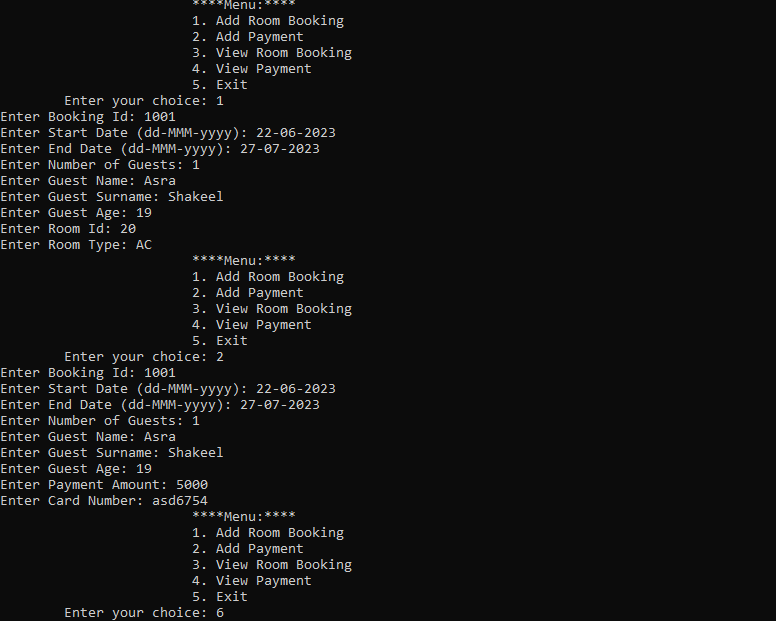
**Code:**







**Output:**



**Working:**

**Guest Class:**

* **Attributes:** name, surname, age
* **Methods:**
  + setGuest(string name, string surname, int age): Sets the attributes of a guest.
  + getGuest() const: Prints the attributes of the guest. The const keyword indicates that this method does not modify any member variables.

**Booking Class:**

* **Attributes:** id, from, to, guest\_no, guest
* **Methods:**
  + setBooking(int id, string from, string to, int guest\_no, Guest guest): Sets the booking details.
  + getBooking() const: Prints the booking details, including guest information.

**Room Class (inherits from Booking):**

* **Attributes:** room\_id, room\_type
* **Methods:**
  + setRoom(int room\_id, string room\_type): Sets the room details.
  + getRoom() const: Prints the room details.

**Payment Class (inherits from Booking):**

* **Attributes:** amount, cardnumber
* **Methods:**
  + setPayment(int amount, string cardnumber): Sets the payment details.
  + getPayment() const: Prints the payment details.

**Here's a step-by-step explanation of the code:**

**Step 1:** The program starts, and the main function is called.

**Step 2:** The displayMenu function is called, which prints the menu options to the console.

**Step 3:** The user is prompted to enter their choice, and the input is stored in the choice variable.

**Step 4:** The program enters a do-while loop, which will continue to execute until the user chooses to exit (choice 5).

**Step 5:** The program checks the value of choice and executes the corresponding case:

**Case 1:** Add Room Booking

The user is prompted to enter the booking ID, start date, end date, number of guests, guest name, surname, age, room ID, and room type.

The guest object is created and its attributes are set using the setGuest method.

The room object is created and its attributes are set using the setBooking and setRoom methods.

**Case 2:** Add Payment

The user is prompted to enter the booking ID, start date, end date, number of guests, guest name, surname, age, payment amount, and card number.

The guest object is created and its attributes are set using the setGuest method.

The payment object is created and its attributes are set using the setBooking and setPayment methods.

**Case 3:** View Room Booking

The room object's getBooking and getRoom methods are called to display the booking and room details.

**Case 4:** View Payment

The payment object's getBooking and getPayment methods are called to display the booking and payment details.

**Case 5:** Exit

The program prints a message indicating that it is exiting.

**Default:** If the user enters an invalid choice, an error message is printed.

**Step 6:** The program loops back to the displayMenu function and prompts the user to enter their choice again.

This process continues until the user chooses to exit (choice 5).

**Conclusion:**

In conclusion, this C++ code demonstrates a basic implementation of a hotel management system using object-oriented programming (OOP) concepts. The code defines four classes - Booking, Room, Guest, and Payment - each representing a distinct entity in the hotel management system. The classes have constructors to initialize attributes and member functions to display entity details.

The main function creates two sets of objects for each class and calls the member functions to display the details of each object. This code showcases the fundamental principles of OOP, including classes, inheritance, and constructors.

**The end**