Simple Hotel Management System – OOP Task

Objective:

Implement a simple hotel management system using the concepts you've learned:

- Classes and Objects
- Fields (attributes)
- Properties (auto, read-only, write-only, with validation)
- Methods (instance, static, getters/setters)
- Constructors
- Encapsulation (using private fields and controlled access)

Requirements:

1. Create a class called Room with the following:
    - Field: roomNumber (int)
    - Field: isBooked (bool)
    - Property to get room number
    - Property to get whether the room is booked
2. Create a class called Guest with the following:
    - Auto-property: Name
    - Auto-property: NationalID
3. In the Room class:
    - Create a method called Book() that sets isBooked to true.
    - Create a method called Free() that sets isBooked to false.
4. In Main method, create one room and one guest, then:
    - Book the room
    - Display room status before and after booking
5. Add constructor overloads to both Room and Guest classes to initialize data during object creation.
6. Add a class Booking with:
    - A constructor that takes a Room and a Guest
    - A method ConfirmBooking() that:
        - Checks if the room is booked
        - If not, books the room and prints guest name and room number

7. Apply encapsulation:
    - Make all fields private
    - Expose access through appropriate get or set accessors
8. Add validation:
    - Room number must be positive
    - Guest name and ID cannot be empty
    - Booking can't be done twice for the same room

9. Add a static field in Room to track the **total number of rooms created**.
    - Create a static method GetRoomCount() that returns the count.
10. Add a **read-only property** BookingTime to the Booking class that stores the time of booking when ConfirmBooking() is called.
11. Add a **write-only property** Password in Guest class for future account access (value should be stored in a private field).
12. Add validation inside property setters:
    - Guest name must be at least 3 characters
    - Room number cannot be below 100
13. Create a static class HotelUtils with:
    - A static method PrintWelcomeMessage() that prints a welcome message to the guest
    - A static method IsRoomAvailable(Room room) that returns true/false

Submission Instructions:

- You must use **only the concepts we've covered so far**.
- Avoid using List, Inheritance, Interfaces, or any other advanced topic.
- Include a short comment above each class and method to describe its purpose.