Homework 1

🏨 Hotel Room Booking System (OOP Project)

🎯 Objective:

Design a **Hotel Room Booking System** using **Object-Oriented Programming (OOP)** principles. The system will manage rooms, customers, and bookings while applying key OOP concepts like **inheritance**, **encapsulation**, **polymorphism**, and **composition**.

📌 Core Classes & Structure:

1️⃣ Class: `Room` *(Base Class)*

- **Attributes**:
  - `room_number` (int)
  - `room_type` (str) ➜ e.g., "Single", "Double", "Suite"
  - `price_per_night` (float)
  - `is_booked` (bool)
- **Methods**:
  - `book_room()`
  - `checkout_room()`

- `display_room_info()`

2️⃣ Subclasses for Room Types:

Use **inheritance** to define special room behaviors.

➤ `SingleRoom(Room)`

➤ `DoubleRoom(Room)`

➤ `SuiteRoom(Room)`

- Override methods if needed (e.g., apply special discounts or services).

3️⃣ Class: `Customer`

- **Attributes**:
  - `customer_id` (int)
  - `name` (str)
  - `contact_info` (str)
- **Methods**:
  - `display_customer_info()`

4️⃣ Class: `Booking`

- Represents a booking record.
- **Attributes**:
  - `booking_id` (int)
  - `customer` (Customer object)
  - `room` (Room object)
  - `nights` (int)

- `total_price` (calculated)

- **Methods**:
  - `display_booking_info()`

5️⃣ Class: `Hotel`

- Manages all rooms and bookings.
- **Attributes**:
  - List of `Room` objects
  - List of `Booking` objects
- **Methods**:
  - `add_room()`
  - `list_available_rooms()`
  - `make_booking(customer, room_number, nights)`
  - `checkout(booking_id)`

✨ Features to Implement:

- **Add Rooms to Hotel**
- **Register Customers**
- **List Available Rooms**
- **Book a Room**
- **Checkout and Free the Room**
- **View Booking Details**

Main

```
=== Hotel Room Booking System ===
1) Add Room
2) Register Customer
3) View Available Rooms
4) Make Booking
5) Checkout
6) View All Bookings
7) Exit
```

LY RATANA