OOPS. (object Oriented Programming).

$$\begin{tikzcd} & \boxed{\text{object}} \arrow[dl] \arrow[dr] & \\ \text{Properties} & & \text{Behaviour (Method)} \end{tikzcd}$$

*..

Principles

Encapsulatⁿ. Abstractⁿ, Polymorphᵐ, Inheritance.

Encapsulatⁿ → Bundling of the data & method into Single unit → (Class).

Eg - Hiding data & making it a private Var.

For example in a theatre, it has its own Shows, it doesn't mix with others.

Abstraction :- It shows only what is important and hide extra details.

For example - while payment it does not show the entire bank process it only shows success after the transaction is done.

Polymorphism - One action, many results.

"Pay" button can be UPI / Card / Wallet - all behave differently but have same purpose.

Inheritance - Reusing things ; if two people Share Common features (user & admin) we don't write them twice .

Class - user defined blueprint from which obj are created. We can create multiple obj with same behaviour using class .

*. Method (Behaviour) .

Real World Examples

Make my Trip :

In the app we see things like users, hotels, bookings. Each of this can be class.

Class
$\boxed{\text{Hotel}}$
↓
Hotel Name
Location
No. of rooms
} Data

$\boxed{\text{Actions}}$
↓
Book a room
Check availability

Class
$\boxed{\text{User}}$
↓
Name
Email
~~Book~~ / ~~Cancel~~
} Data.

$\boxed{\text{Actions}}$
↓
Login
Book / Cancel .

A user open app and book a Hotel, that booking is saved as a booking object and the payment is a payment object.

$$\begin{tikzcd}\boxed{\text{User}} & \boxed{\text{Admin User}} \arrow[l]\end{tikzcd}$$

User
↓
name
email
login()
book()
Cancel()

Admin User
addHotel()
removeHotel()

Hotel
name
Location
bookRoom()

Booking
airline
flightno
bookflight()

Booking
bookingID
User
Confirm()
Cancel().

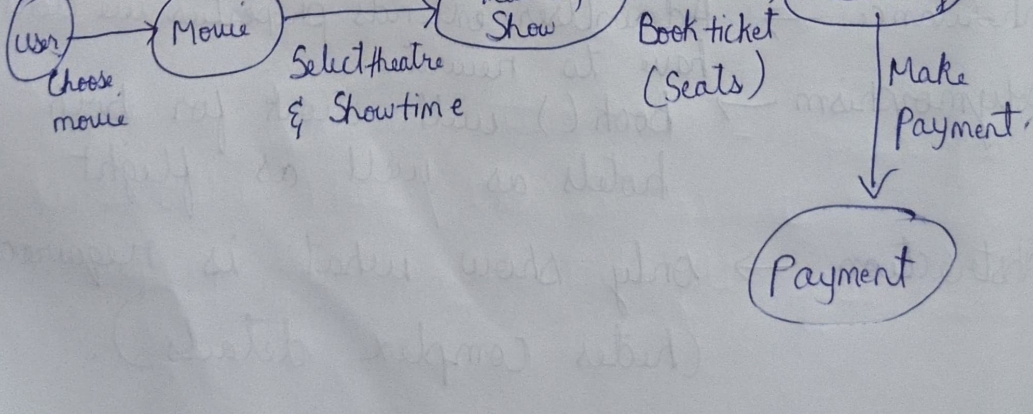
Payment
amount
method
pay()
refund().

Encapsulation → Hide inner details of payment
Inheritance → AdminUser extends properties we dont have to rewrite
polymorphism → book() will work for both hotels as well as flight.
Abstraction → only show what is required (hides complex details).

Book My Show

In BookMyShow we have objects like User, theatre, Movie, Booking & Payment. A user will see a list of movie playing at different theatre. Each theatre has many shows at different timings. When user books ticket for a show, a Booking object is created with user details, show name & seat number. The payment object will handle the transactⁿ.

~~With OOP the objects will organ~~
OOP will help to organize these obj with properties (movie name, show time) and methods like bookTicket(), cancelTicket(). Encapsulatⁿ will only show the transactⁿ details only and hide the inner details of payment. Inheritance lets admin user to be a special user who can add or remove movies and polymorphism will allow same method bookTicket() to book for movies events or plays.



User
Choose movie
Movie
Select theatre & Showtime
Theatre & Show
Book ticket (Seats)
Booking
Make Payment.
Payment