Features of OOPS (synopsis)

(short synopsis to understand and remember)

`Object`

- Every real-world entity is an object.
- Object is an instance of a class
- Object is a runtime entity of a class definition.
- An object has Behaviour (things it does or performs) and Attributes (things that describe it).
- For eg: A Chair object can have behaviour like Movement, Height Adjustment & Attributes like Colour, Make & Model, and Price.

`Class`

- The collection of all related objects is called a class.
- class is a user defined prototype
- class is a definition of an object
- Consider class as a general category which contains all the related objects inside it.
- For eg: Objects like Wheelchair, Office Chair and Wooden Chair can be a part of the "Chair" class.

`Inheritance`

- In general we human beings always know about inheritance.
- In programming it is almost the same. When a class inherits another class it inherits all features (like variables and methods) of the parent class.
- This helps in reusing codes.

- The way we inherited a few qualities from our parents similarly, a class can also inherit the qualities from a parent class.
- For eg: A Phone Class can have two Child Classes: 1) TelePhone and 2) MobilePhone. Both can inherit the "calling" behaviour.

`Encapsulation`

- It means wrapping data into a single unit & securing it.
- For eg: Drug Capsule wraps different medicines into a single unit and protects them from the outside environment.
- Bank Locker wraps your valuables into a single unit(locker) and protects it via passcode.
- `Encapsulation` is seen as the bundling of data with the methods that operate on that data
- The methods for retrieving or accessing the values of attributes are called `getter` methods.
- `getter` methods do not change the values of attributes, they just return the values.
- The methods used for changing the values of attributes are called `setter` methods.

`Abstraction`

- Hiding complexity from the user and showing only the relative stuff.
- For Eg: In Car, all the complexity like the engine, machinery, etc is hidden from you; only relevant parts are shown, like the brakes, accelerator, and gearbox.
- `Information hiding` : Information hiding is the principle that some internal information or data is "hidden", so that it can't be accidentally changed.
- data abstraction is present, if both data hiding and data encapsulation is used.
  `Data Abstraction = Data Encapsulation + Data Hiding`

`Polymorphism`

- It means many forms. With the same name, it provides different forms.
- Python is implicitly polymorphic.
- For eg: In Chess, we've 6 pieces - king, rook, bishop, queen, knight, and pawn. All of them "move" differently i.e. Bishop moves diagonally, Rooks move horizontally and vertically, etc.