Case Studies : Tic Tac Toe
Parking lot
Book My Show
Splitwise.

} LLD-3.

⇒ LLD-1 (language) ⇒ Mock Interview

LLD-2 & 3 ⇒ Mock Interview

Agenda -

⇒ UML Diagrams. → Class Diagram
→ Use Case Diagram.

UML Diagrams.

Communication.

Clients ⇒ Gather requirements

Managers
- Product Managers
- Engineering Managers

} Design Discussions / Approvals.

Business } Requirements.

Ways to communicate?

⇒ Words.

↳ Misunderstanding / Ambiguity.

⇒ Explaining the design using only words isn't good enough because there can be lot of ambiguity.

⇒ Picture.

Pros.

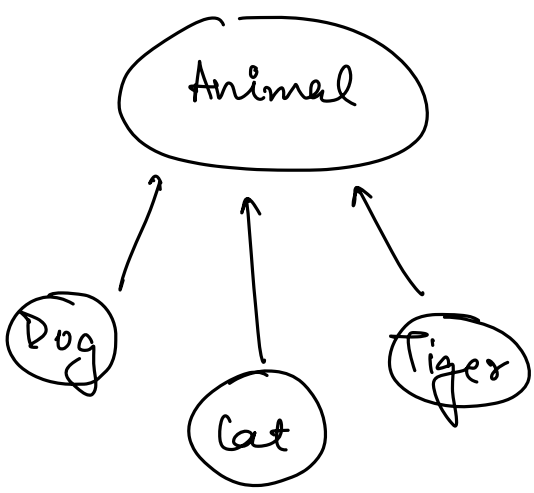
↳ less ambiguous.

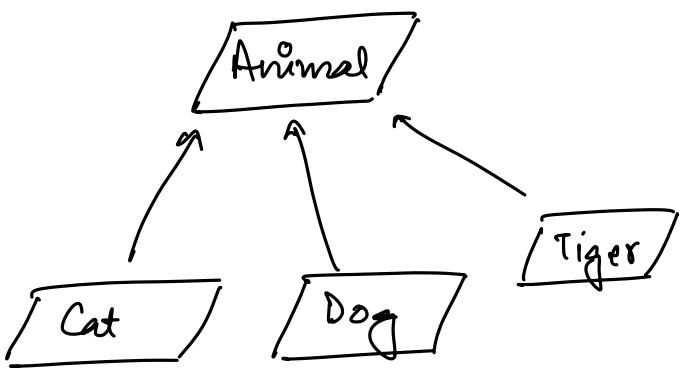
↳ Easy to understand.

↳ Easier to visualize.

Cons.

↳ No Standardization.



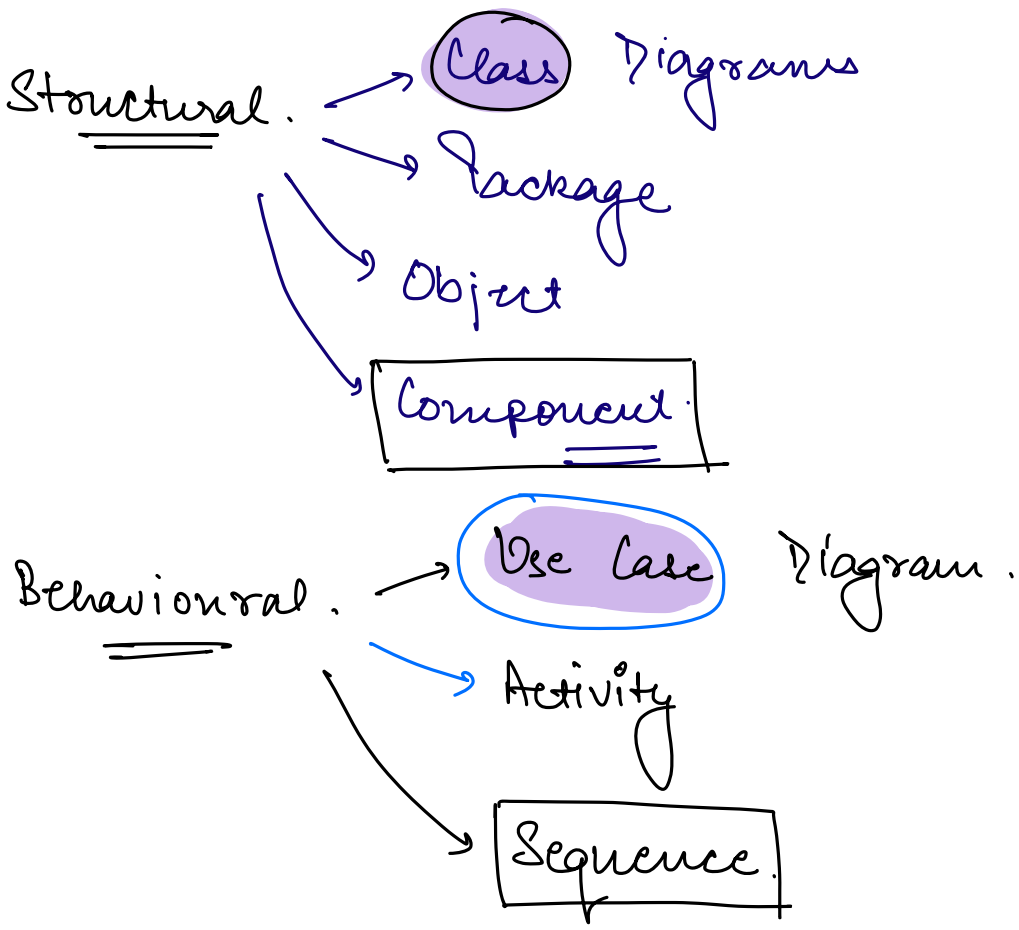


UML (Unified Modeling language).

⇒ Standard way of representing our low level design.

2 type of UML diagrams.

- Structural ⇒ How our codebase looks like?
- Behaviourals.
  - ↳ Methods / Actions.



Use Case

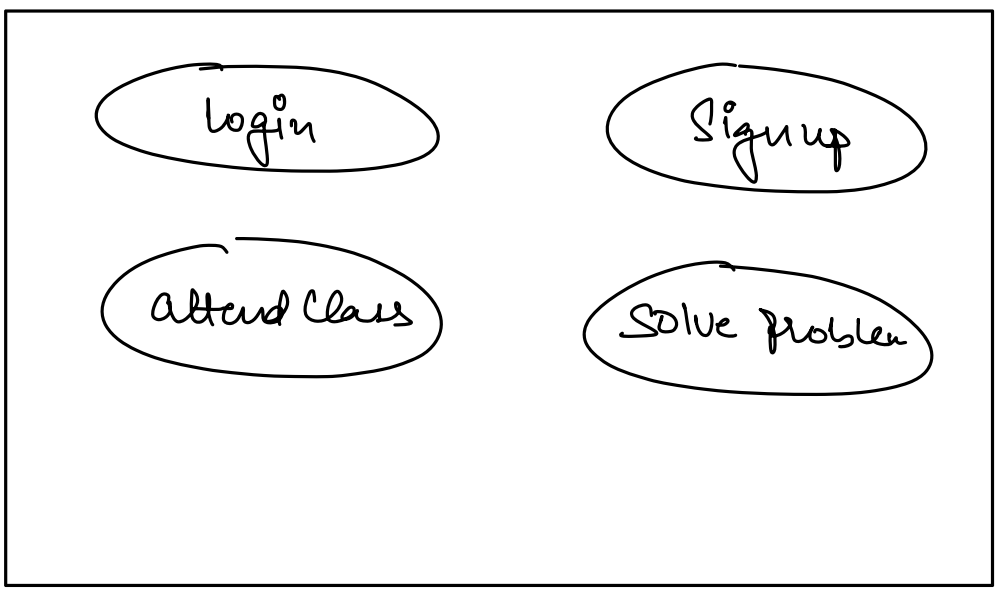
⇒ Behavioural.

⇒ Actions / Methods.

⇒ Different features / functionalities (Use cases.) that are supported by our system, along with their users.

5 Components.

1) System Boundary. ⇒ Represents the scope of the System.



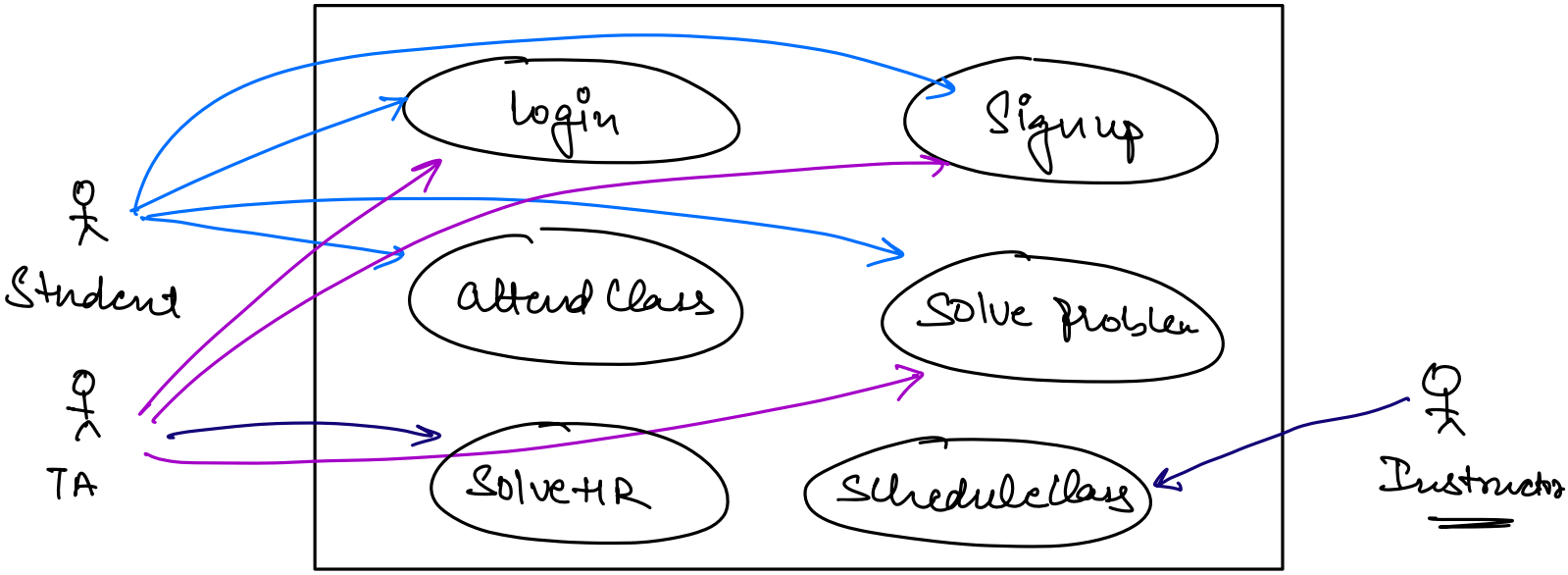
2) Use cases.

- ↳ functionalities / methods.
- ↳ Oval Shape.

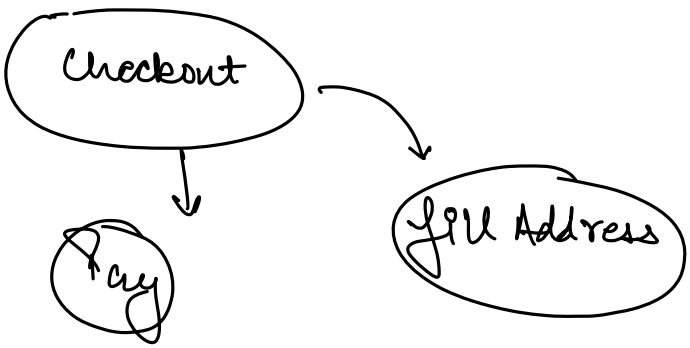
③ Actors.

⇒ Users of the system.

⇒



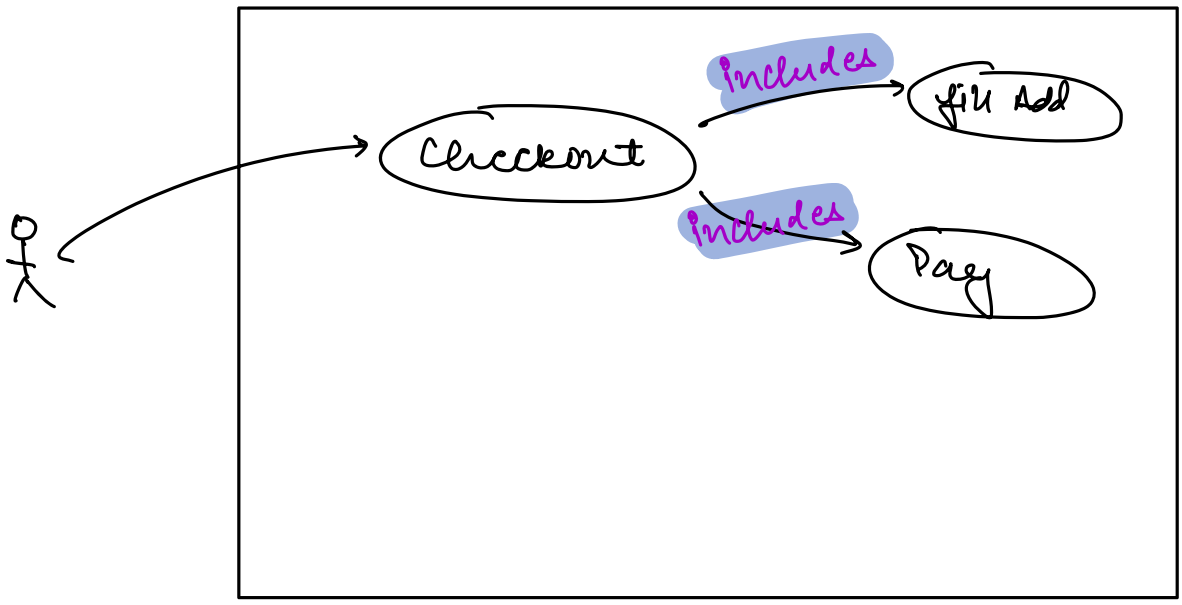
④ Includes.



Checkout includes

→ Pay

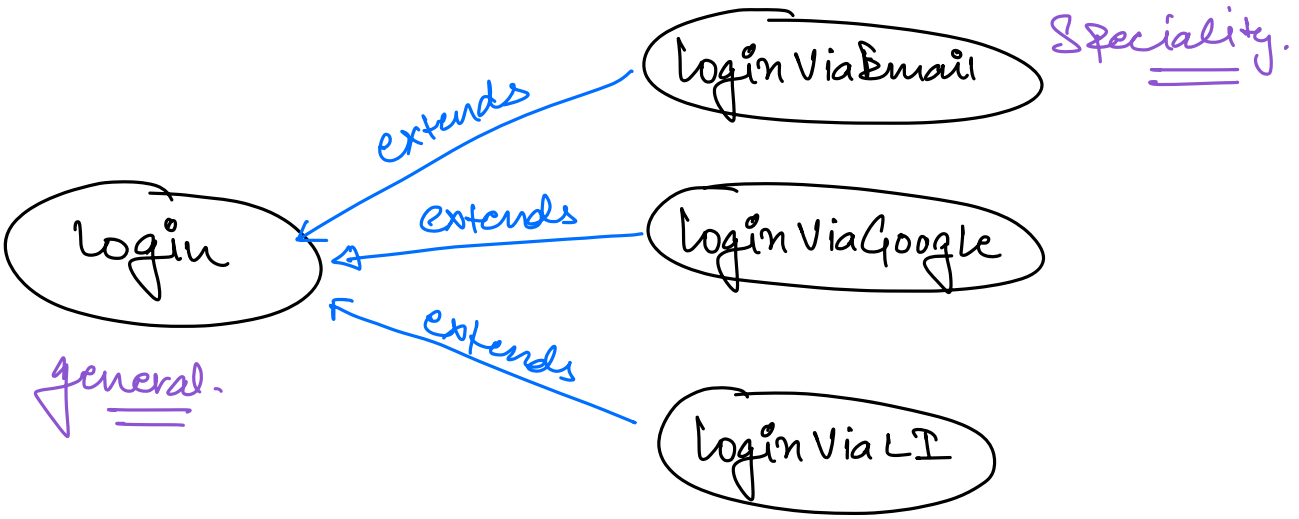
→ fill address.



To complete Checkout, user will have to fill address & pay

(5) Extends.

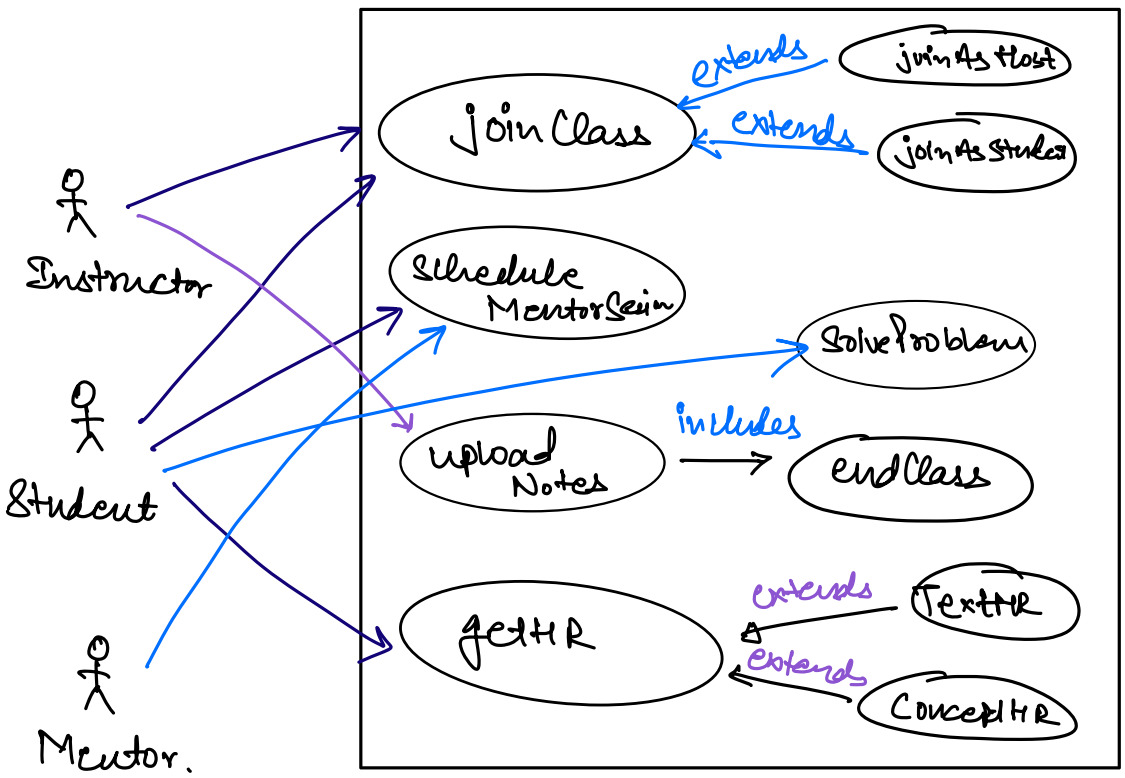
If one feature has multiple variants.



Assignment.

⇒ Draw a use case diagram [SCALER].

1) At least 5 features
2) At least 2 actors.
3) Includes
4) Extends.



Class Diagram.

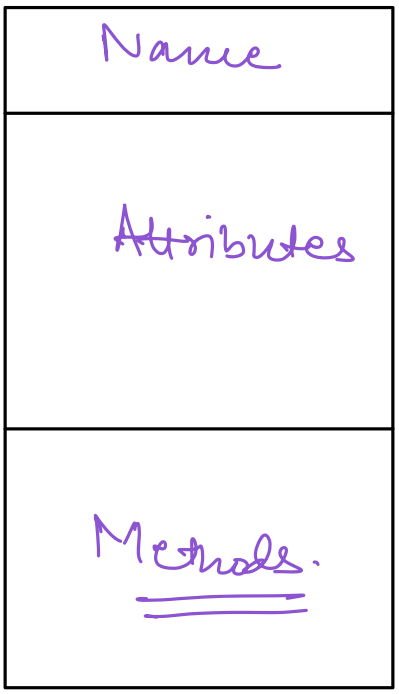
⇒ Used to represent entities in the system & their relations.

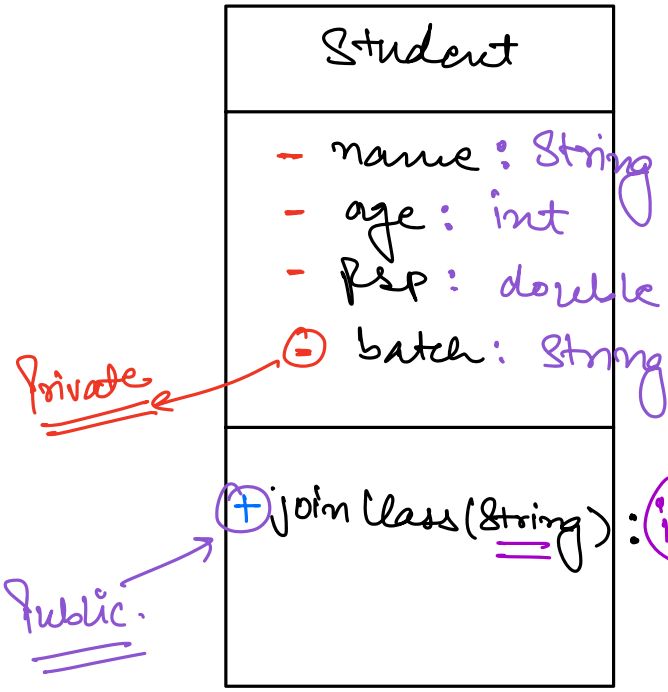
- Class
- Interfaces
- Abstract Class.

⇒ Inheritance relation.

⇒ Class implementing an interface.

⇒ Class





access Modifier — Datatype — attr.

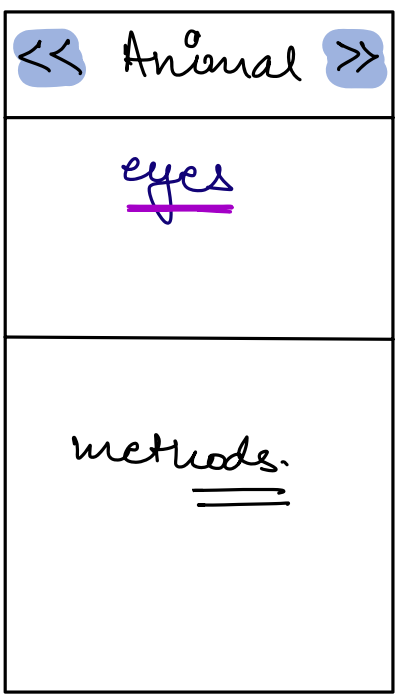
↳ Public : +

↳ Private : −

Protected : #

return type of fun.

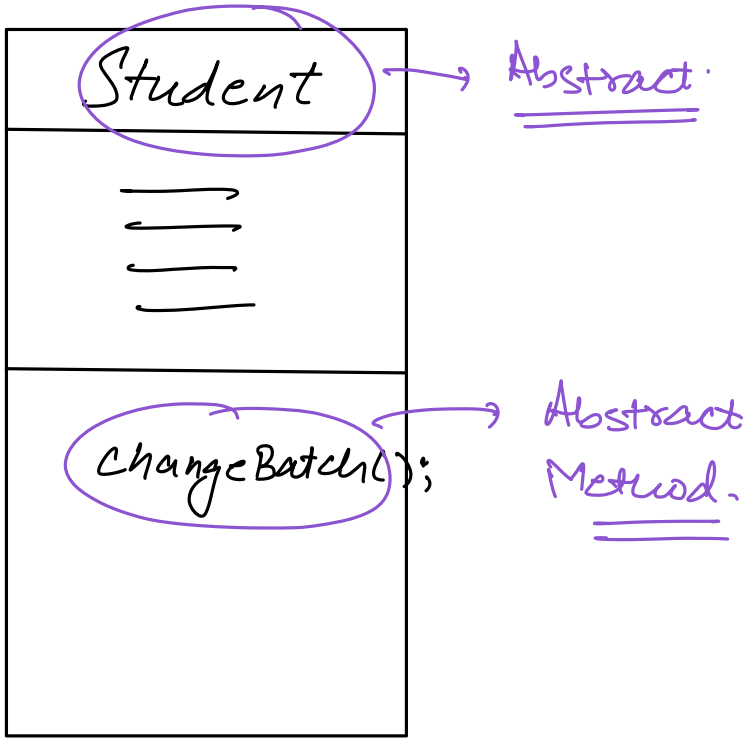
⇒ Interfaces.



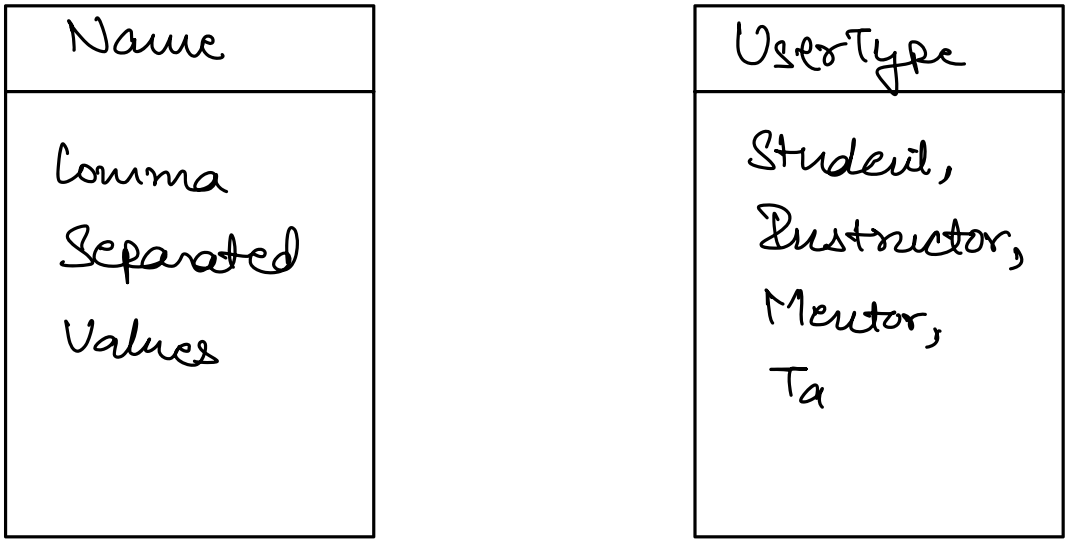
⇒ Static keyword is represented using an underline.

Abstract Class.

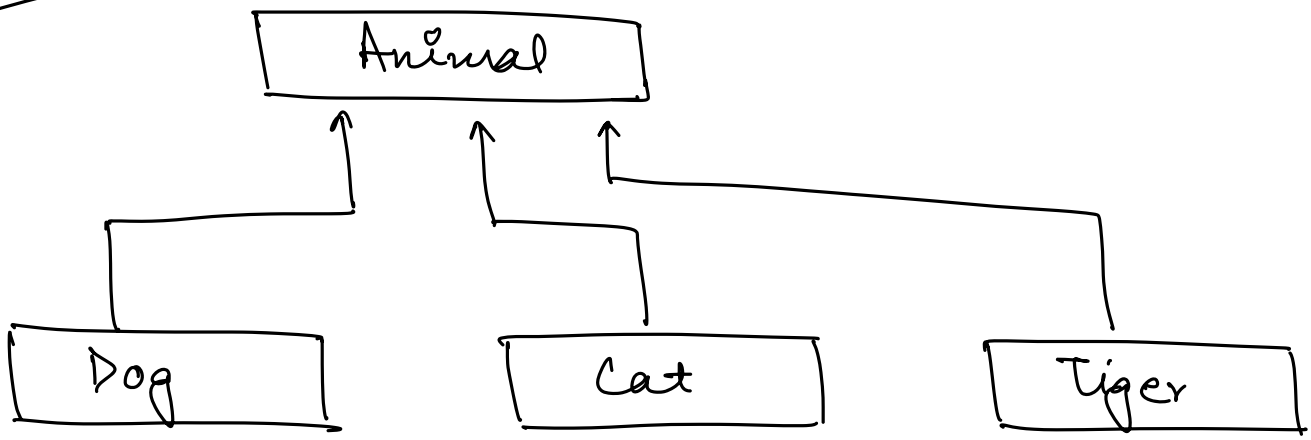
Exactly same as normal class the only diff is to write the class name in italics.

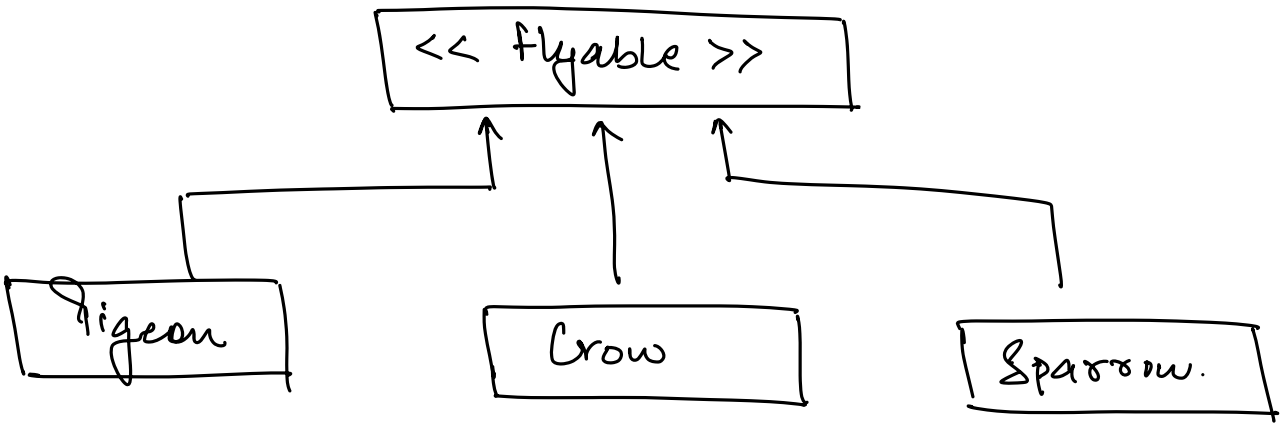


Enum

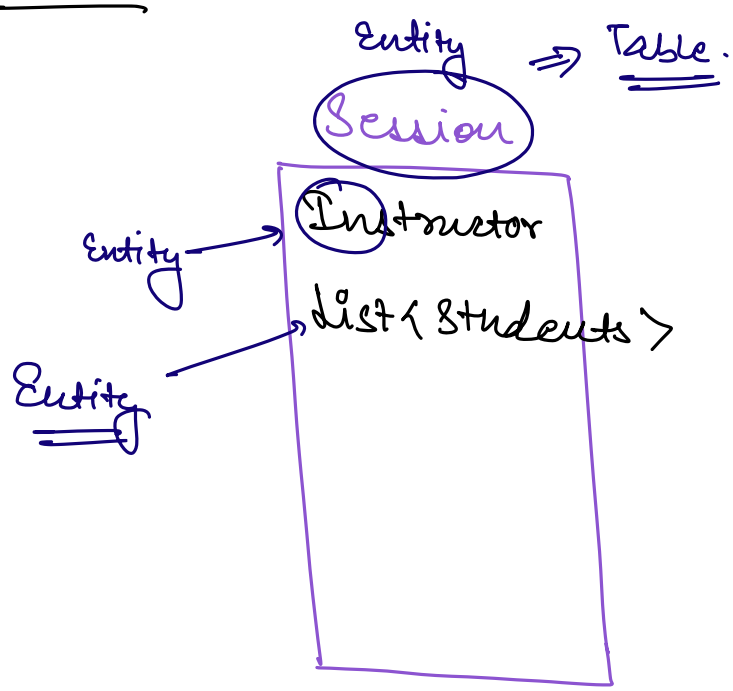


IS-A.

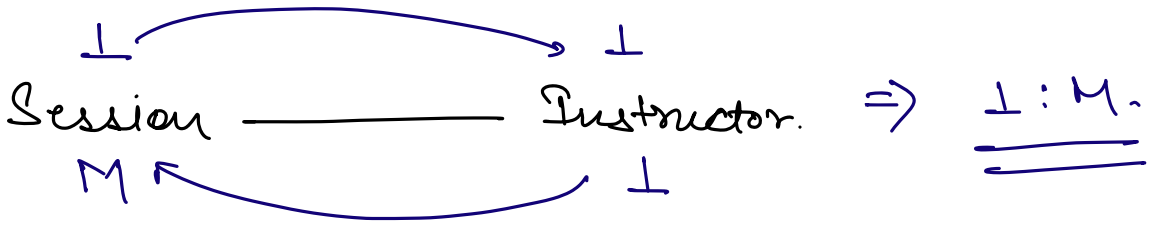




HAS-A.



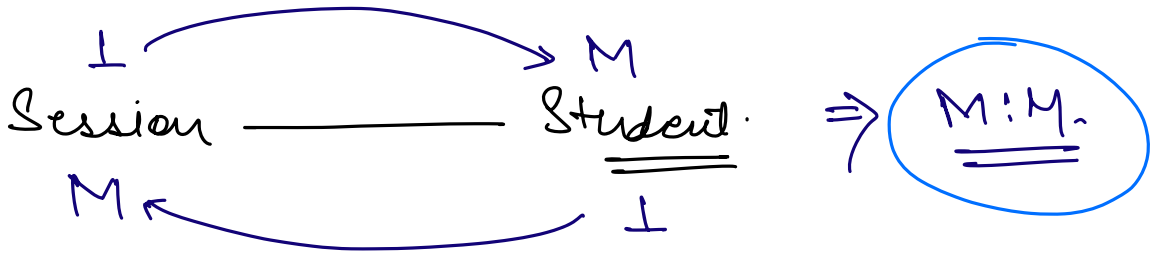
Session HAS-A Instructor



⇒ 1 : M.

⇒ Id of (1) side on (M) side.

⇒ Session table will have instructor id.



⇒ M : M.

⇒ Mapping table.

Student_sessions

St_id	Session_id

———— * ————

LLD-3.