Class Diagram:-

Attributes:-

A significant piece of data containing values that describe each instance of that class.

Operations:-

Allow you to specify any behavioral feature of class

- private:-

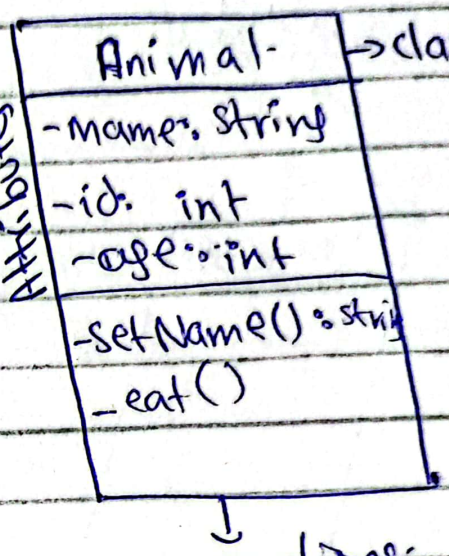
can't access by any class or subclass.

+ public:-

can be access.

protected:-

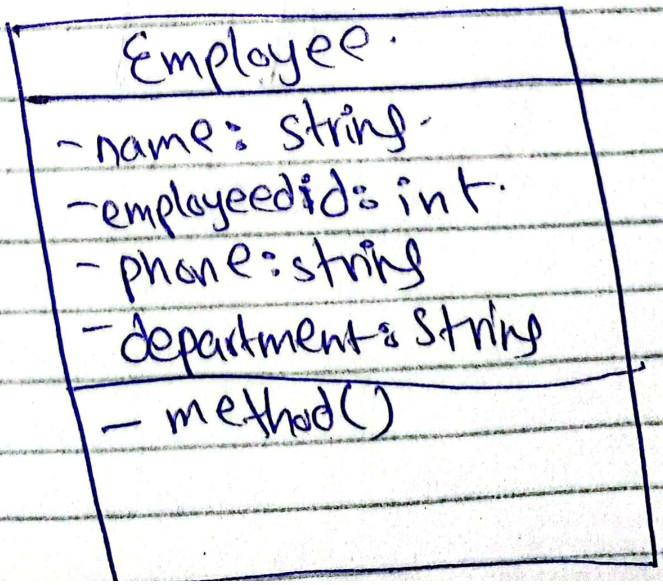
These can be accessed by the same class or its sub-classes.



Visibility -

- private
+ public
# protected.

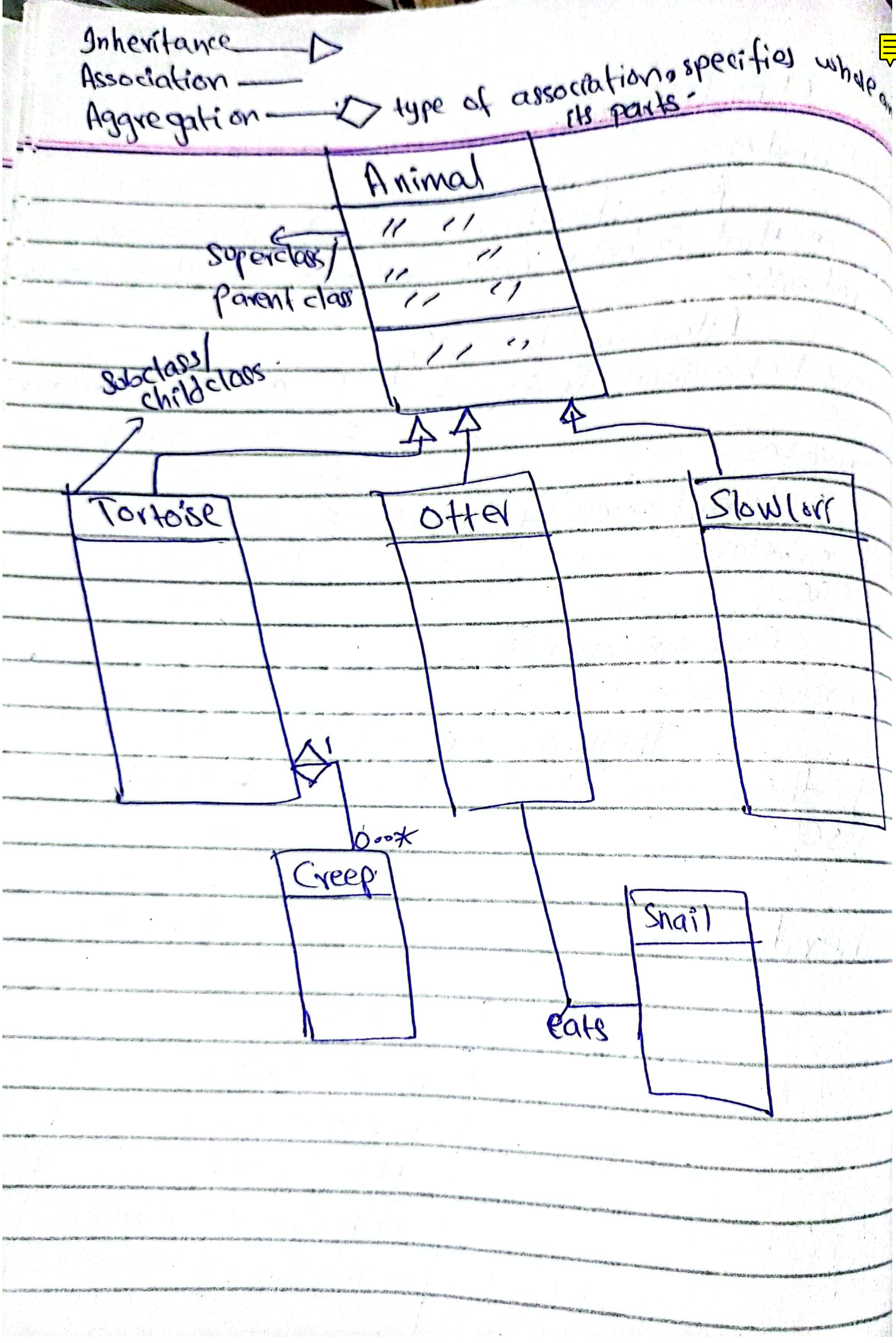
Inheritance:-



Inheritance ——▷

Association ——

Aggregation ——◇ type of association, specifies whole and its parts.



Aggregations-

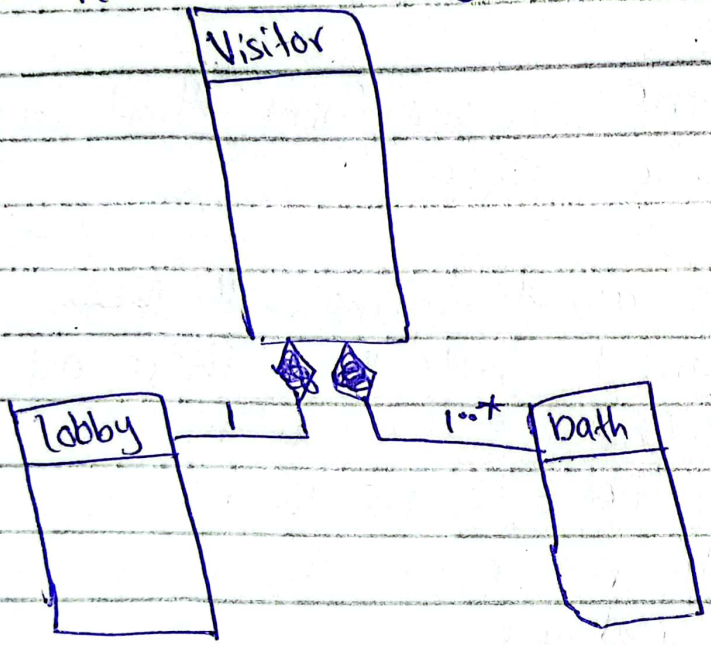
where a part can exist outside the whole is aggregation

Creep is a group of tortoise, if a tortoise ~~is~~ be separated from creep it this can exist on its own.

Compositions-

Part can't exist outside the whole.

Centre has one lobby or atleast one bathroom.



Multiplicity :-

0...1 (zero to one) optional

n specific number

0...* Zero to many

1...* One to many.

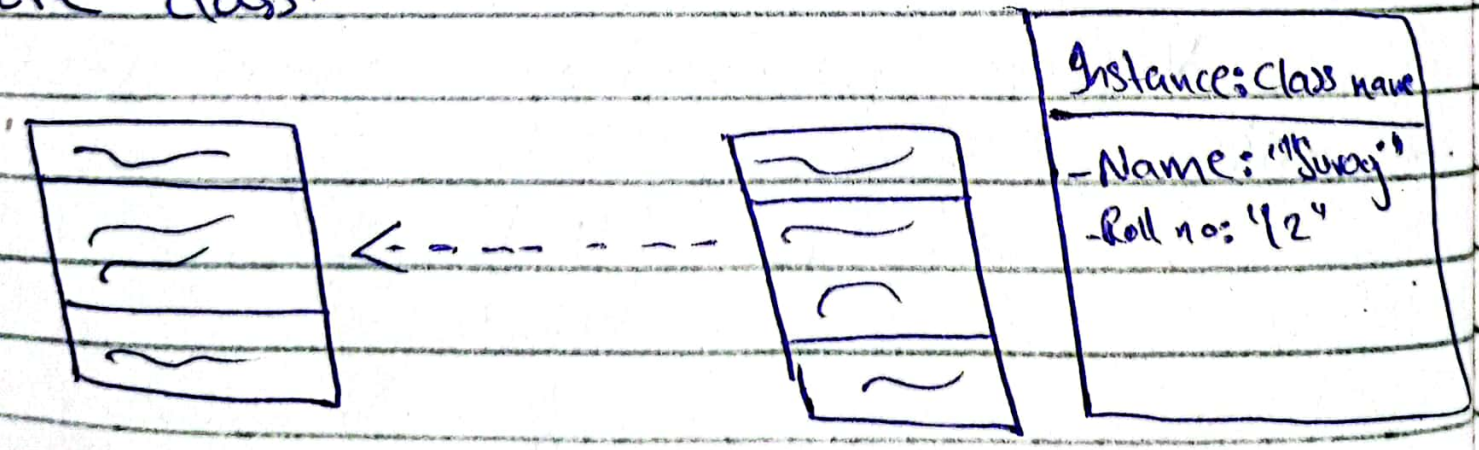
m...n specific number range.

Object diagram:-
It is snapshot of the objects in the system:-
Shows instances of class diagrams Eg links among them.

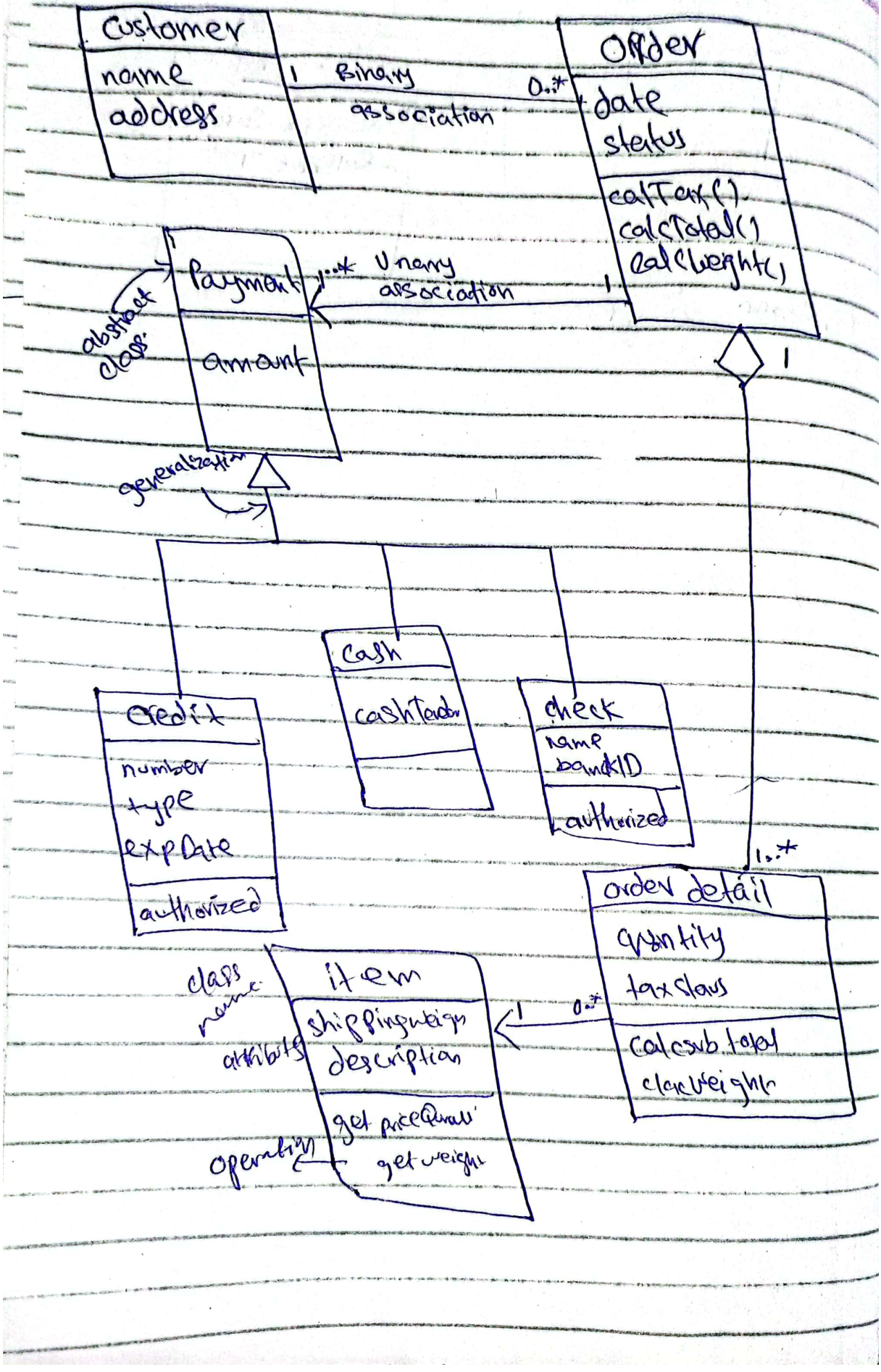
How classes can communicate?
Synchronous: ⟶
Synchronous message requires a response before the interaction can continue.

Asynchronous: ⟶ Eg ⟶
Don't need a reply for interaction to continue.

Reply:- ⟵ - - - -
When a class replies back to the sending one class.



Class Diagram:-

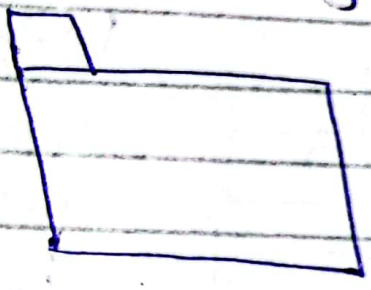


Package diagram:-

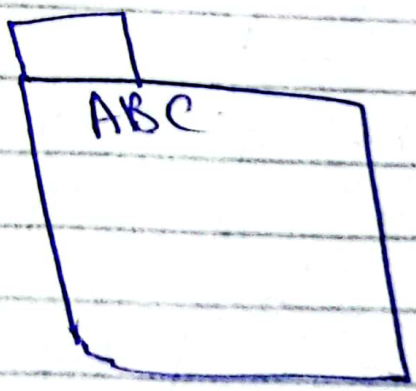
To organize complex diagrams, you can group classes into packages. It is the collection of logically related UML elements.

- Notation:-

→ Packages diagram appears as rectangles with small tabs at the top.

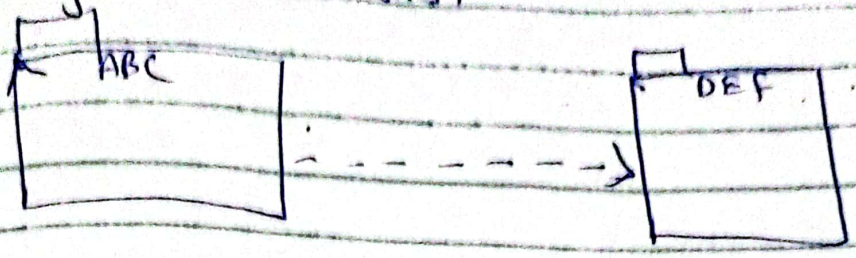


→ The package name is on the tab or inside the rectangle.

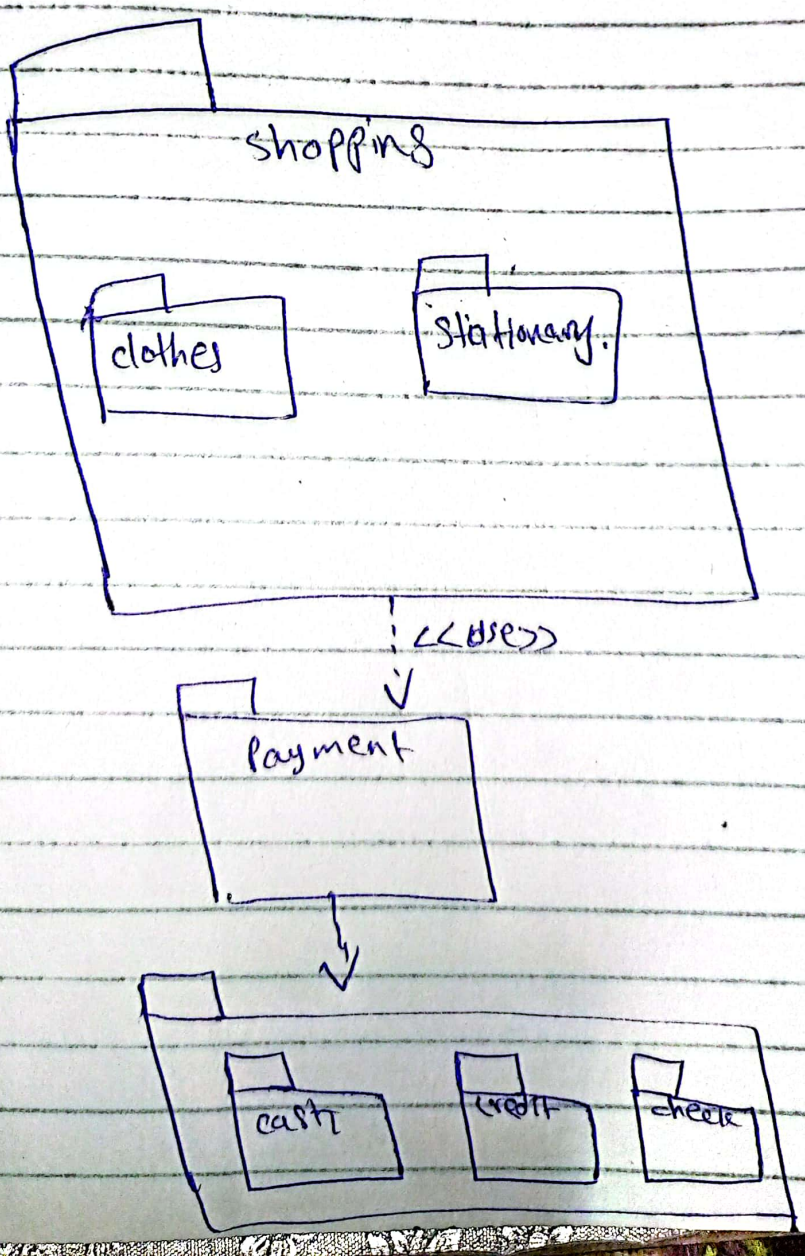


→ The dotted arrows are dependencies. One package depends on another if changes in the other could possibly force

changes in the first

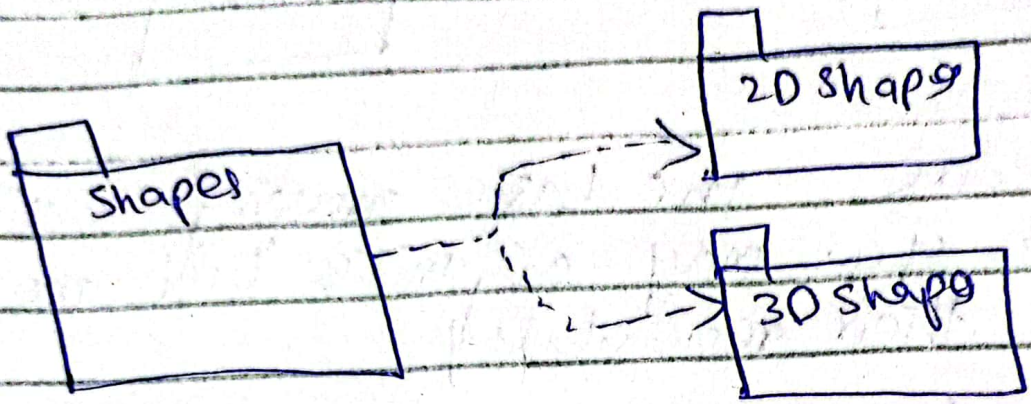


→ Packages are the basic grouping construct with which you may organize UML models to increase their readability.
→ Dynamic.
→ Packages contain more than one package



<< Import >>

Indicates that functionality has been imported from one package to another.



-<<access>>.

Indicates that one package requires assistance from the functions of another package.