

Agenda.

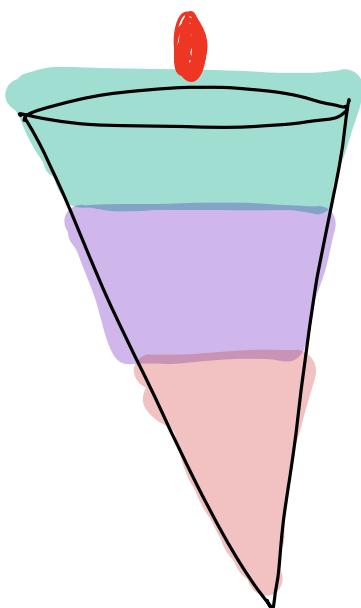
- Decorator
- Flyweight.

⇒ DECORATOR PATTERN.

- ↳ Icecream Parlour. | Pizza | Coffee MIC.

⇒ ICECREAM.

- ↳ Custom ice creams.



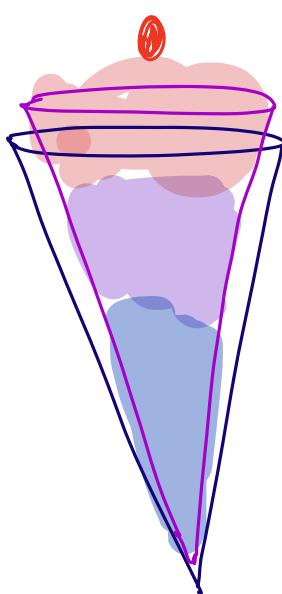
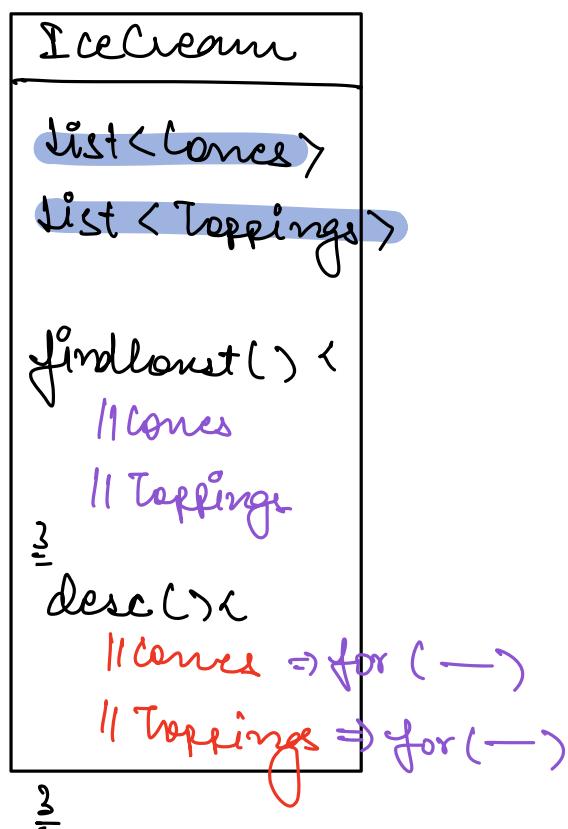
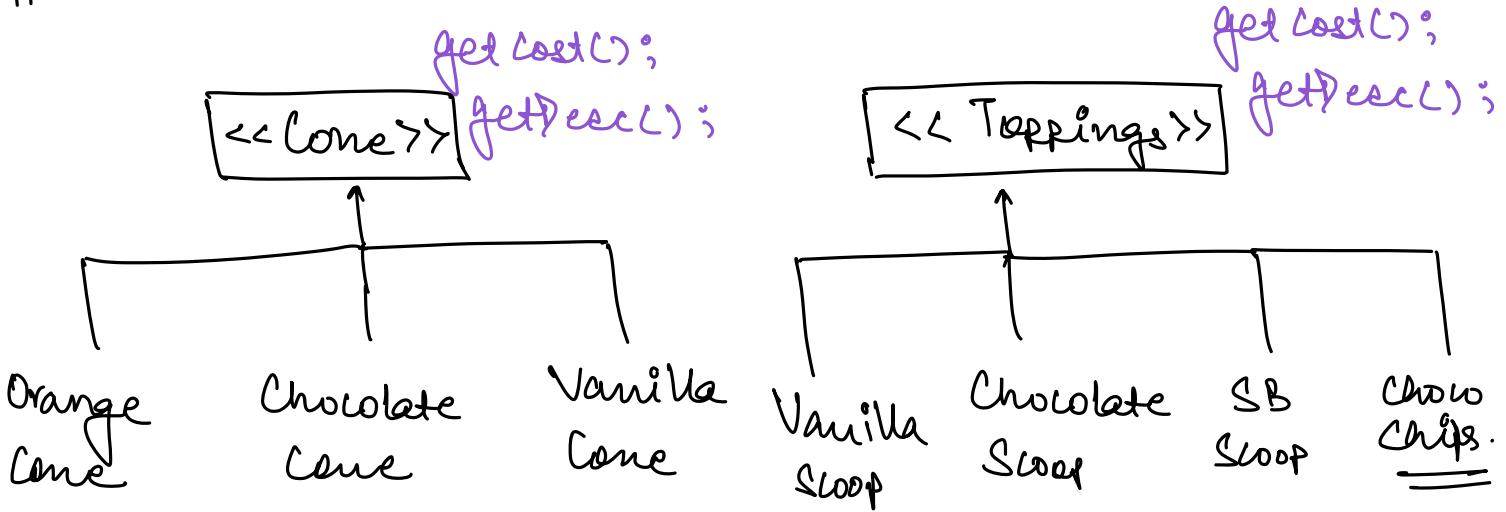
- App" that takes orders for ice creams.
- Custom config.

⇒ Build ice cream.

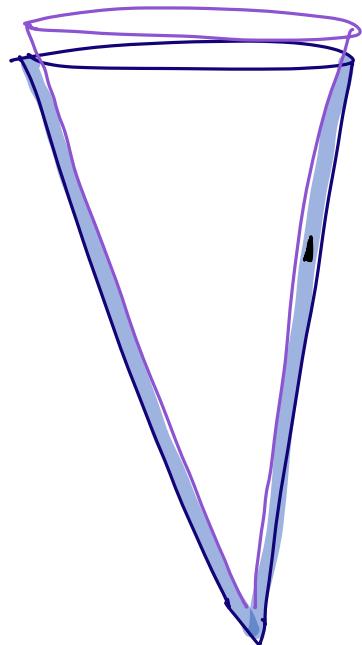
- ⇒ get the cost of the ice cream.
- ⇒ get Description of the ice cream.

}

#

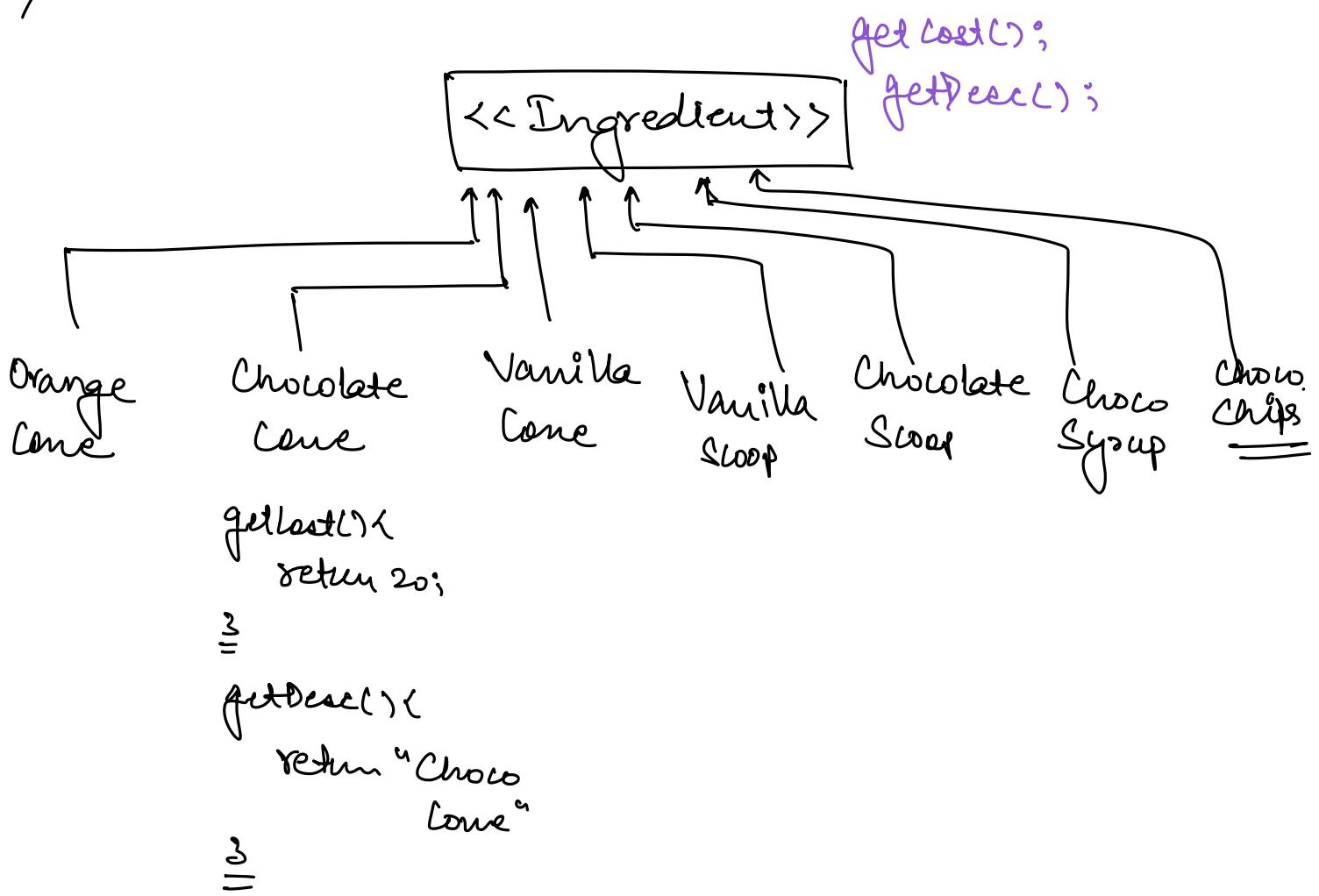


⇒

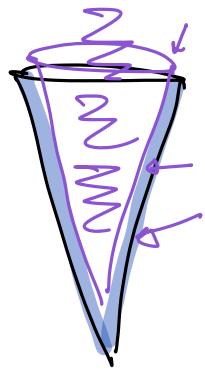


⇒ Sequence of ingredients
can't be managed.

⇒

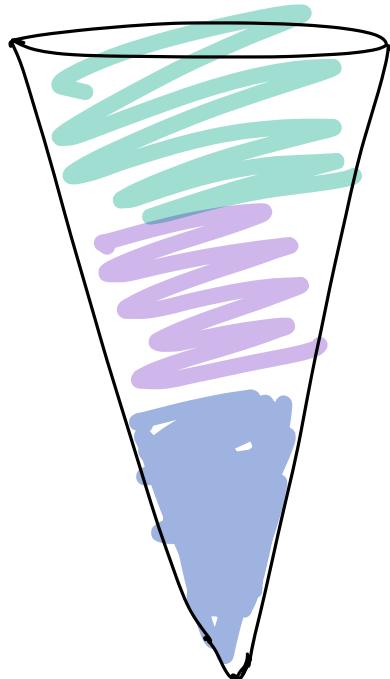


IceCream
 List<Ingredient>
 getCost()<
 for (Ingredient) <
 cost += ing.getCost();
 }
 getDesc()<
 for (Ingredient) <
 desc += ing.getDesc();
 }

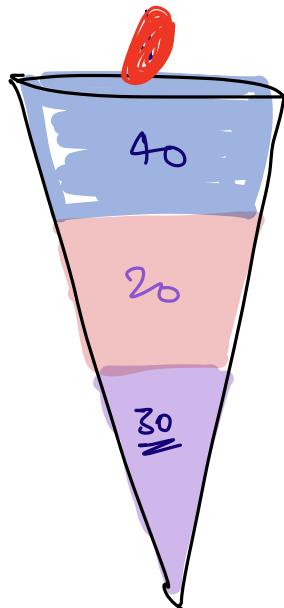


ingredients.add(OrangeCone);
 ingredients.add(ChocoSyrup);
 ingredients.add(ChocoCone);

Decorator.



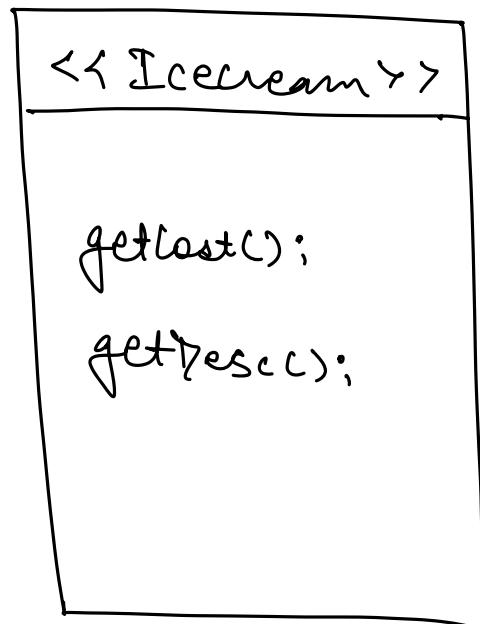
⇒ Right from the beginning we had an ice cream, & then we can keep on adding multiple ingredients on top of it.



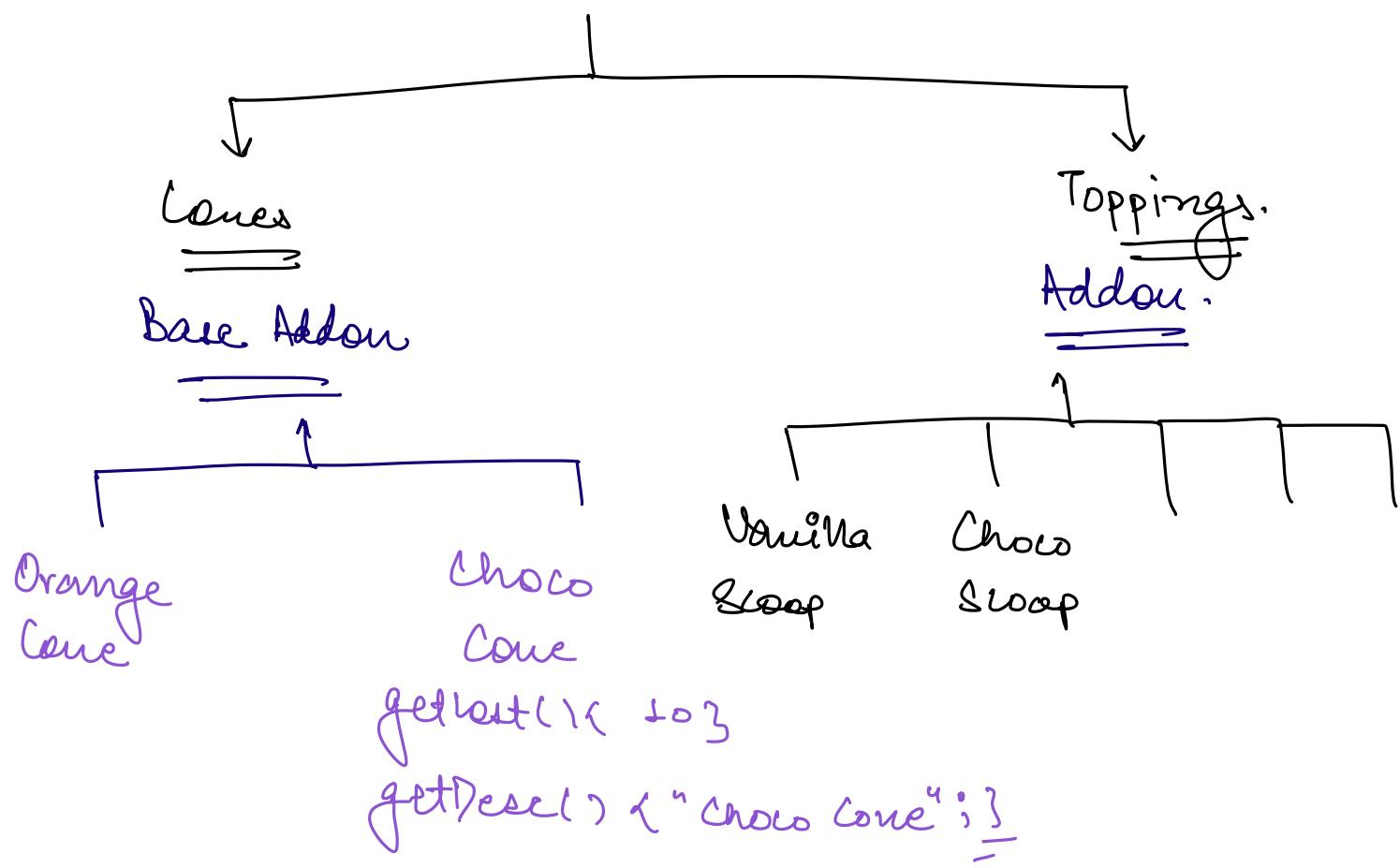
⇒ 20 lbs. + 30 lbs. + 20 + 40 + 5
"cone" + "Scoop-1" + "SB Scoop"
+ "BB Scoop" + "cherry"

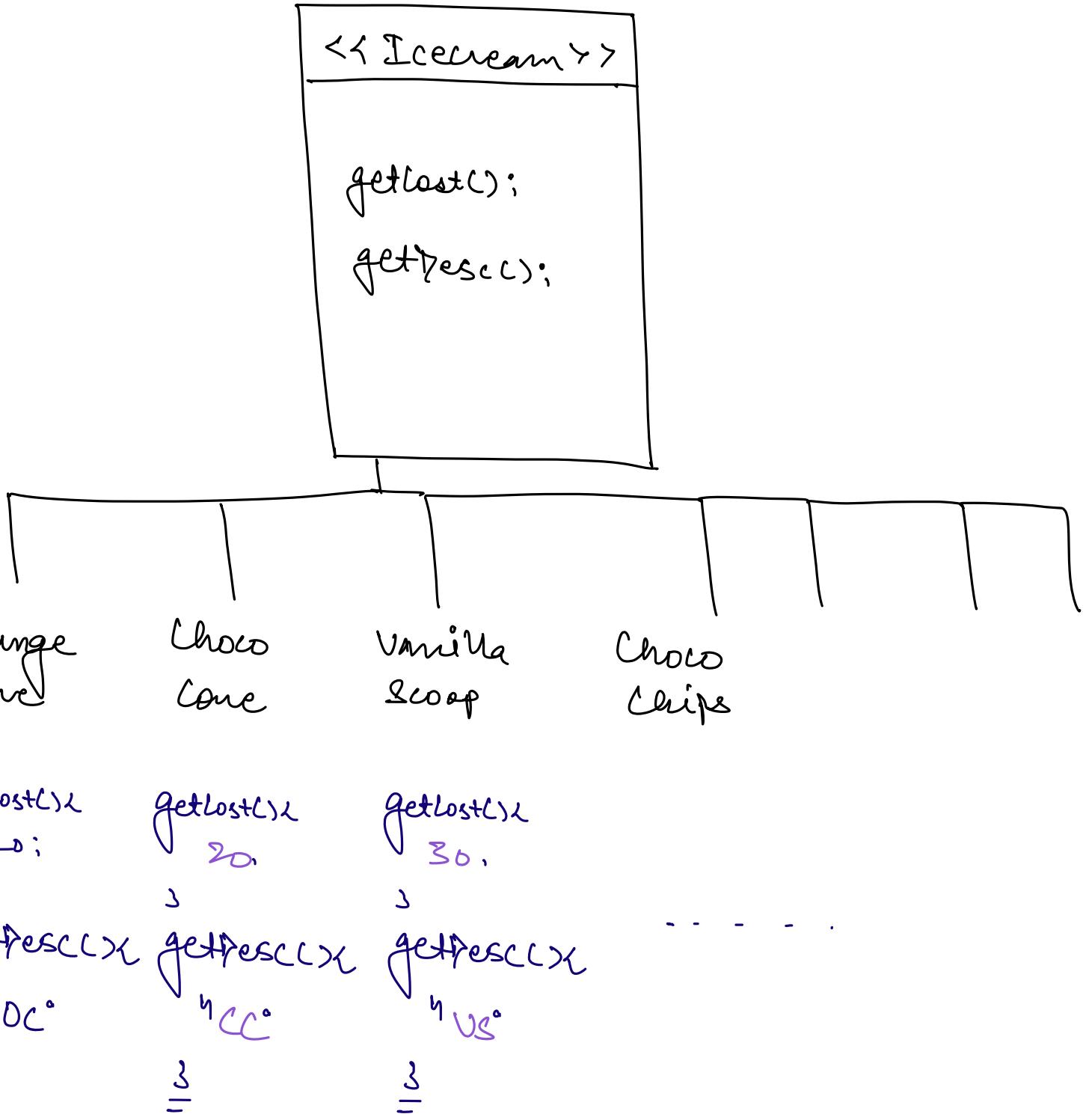
Step 1 :

Create an interface that is going to represent the entity that we are constructing.



Step 2: There are 2 kind of ingredients.





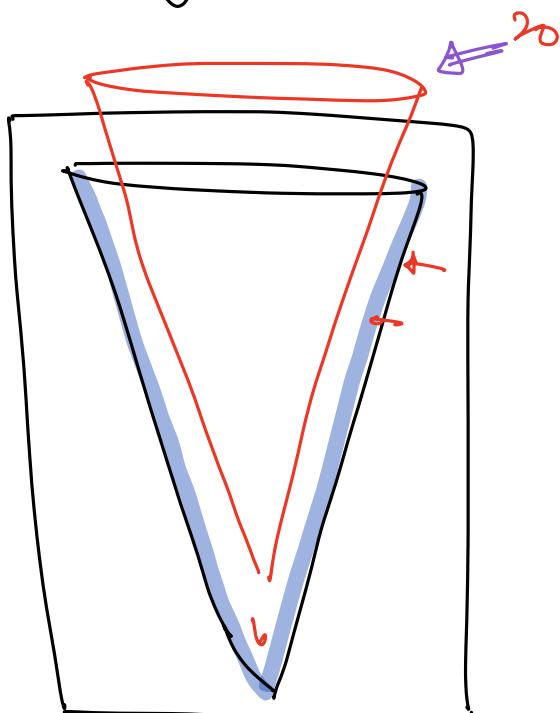
Base AddOn \Rightarrow can be the first item in the ice cream \oplus can also be added as an addon on an existing ice cream.

① Base Addon is the first component in the Re.

`getCost()` \Rightarrow Cost of the item itself.

`getDesc` \Rightarrow Name of the item itself.

② Base Addon, we are adding on top of an existing icecream.



$$\text{Cost} = \underline{\underline{\text{getCost}() + 20.}}$$

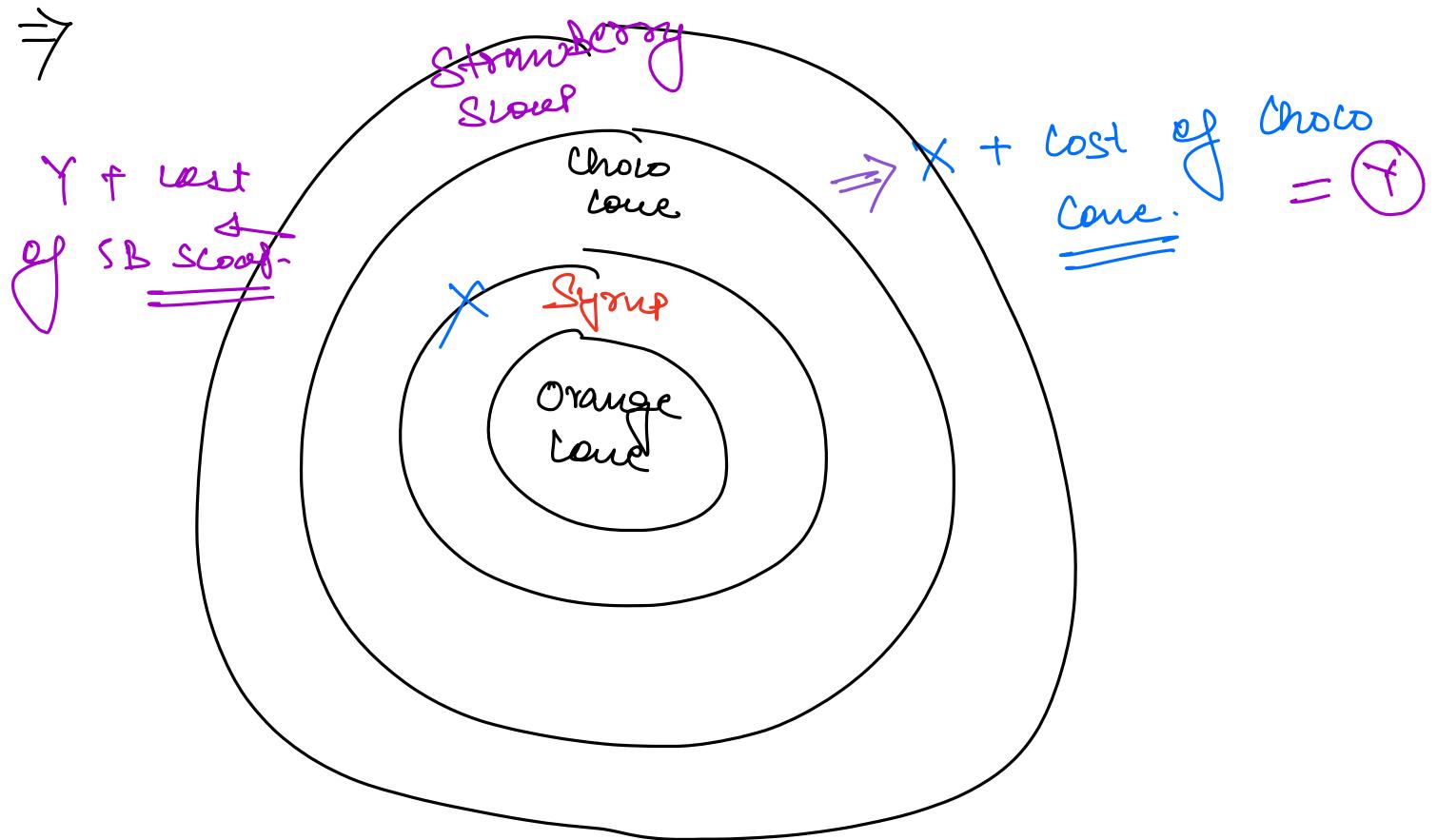
$$\text{Desc} = \underline{\underline{\text{getDesc}() + "OC".}}$$

Addon.

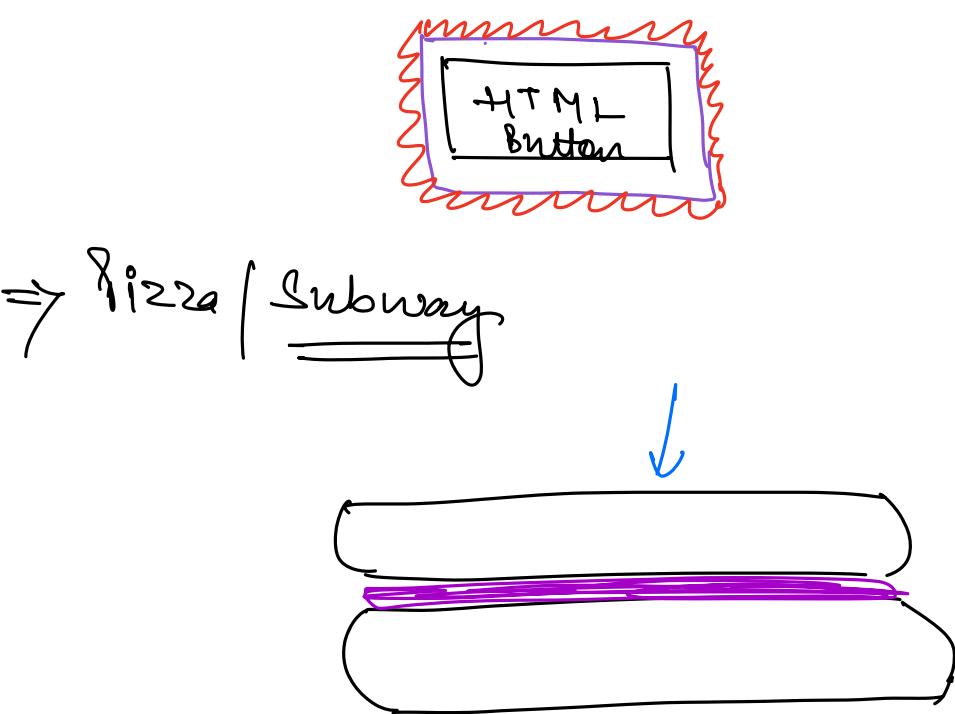
we are adding on top of an existing icecream.

$$\text{Cost} = \underline{\underline{\text{getCost}() + 20.}}$$

$$\text{Desc} = \underline{\underline{\text{getDesc}() + "OC".}}$$

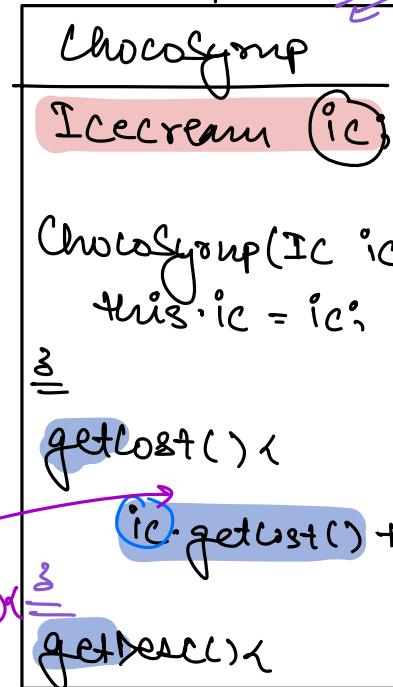
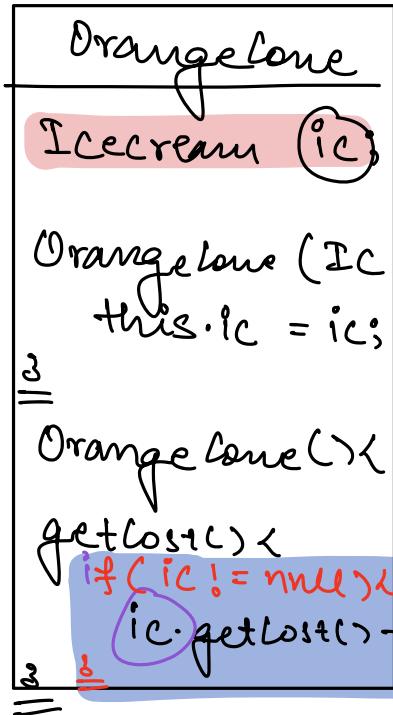


⇒



`<< Icecream >>`

`getPecC();
getCostC();`



`Icecream ic =`

`new VanillaScoop`

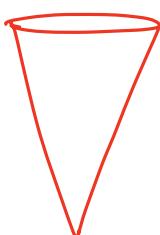
`new ChocoLane (`

`new ChocoSyrup (`

`new OrangeLane () IC`

$\Rightarrow IC$

$\Rightarrow IC$



`);`

⇒

$\text{ic} \cdot \text{getCost}() \Rightarrow 95$

↳ $\text{vanillaScoop} \cdot \text{getCost}()$



$\boxed{\text{Chocolcone} \cdot \text{getCost}()} + 50;$



$\text{ChocolateScoop} + 20;$



$\text{OC} \cdot \text{getCost}() + 15$



10

Icecream $\text{ic} = \boxed{\text{new OrangeIceC}};$

$\text{ic} = \boxed{\text{new ChocoScoop}(\text{ic})}$

$\text{ic} = \boxed{\text{new Chocolcone}(\text{ic})};$

$\text{ic} = \text{new VanillaScoop}(\text{ic});$

Decorator

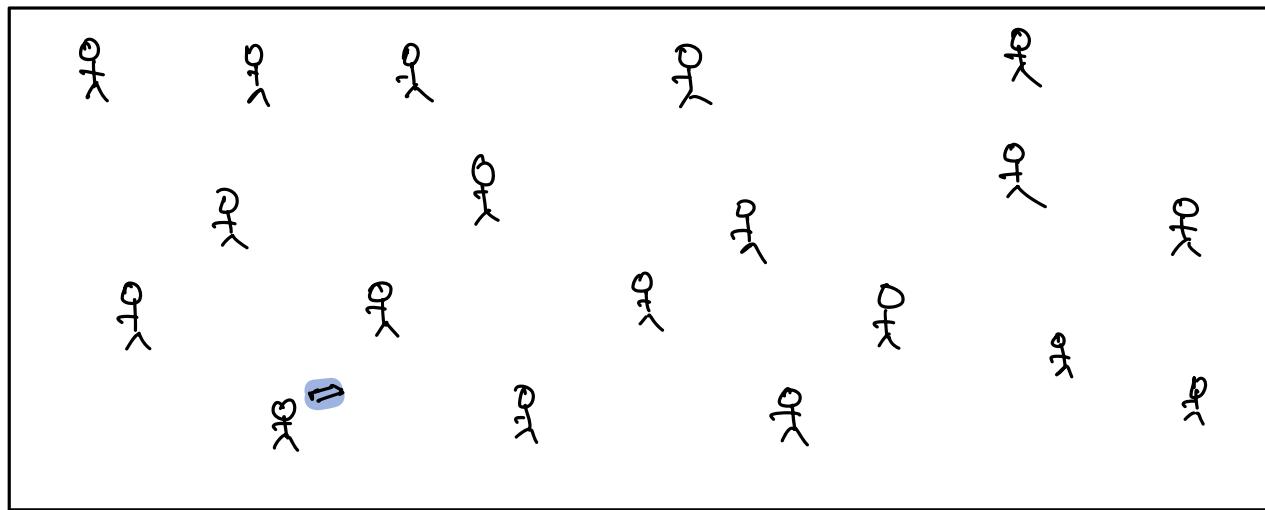
⇒ When we want to keep adding the properties at run time.

FLYWEIGHT.

⇒ Building some game application.

↓
Multiplayer.

Eg: Pubg.

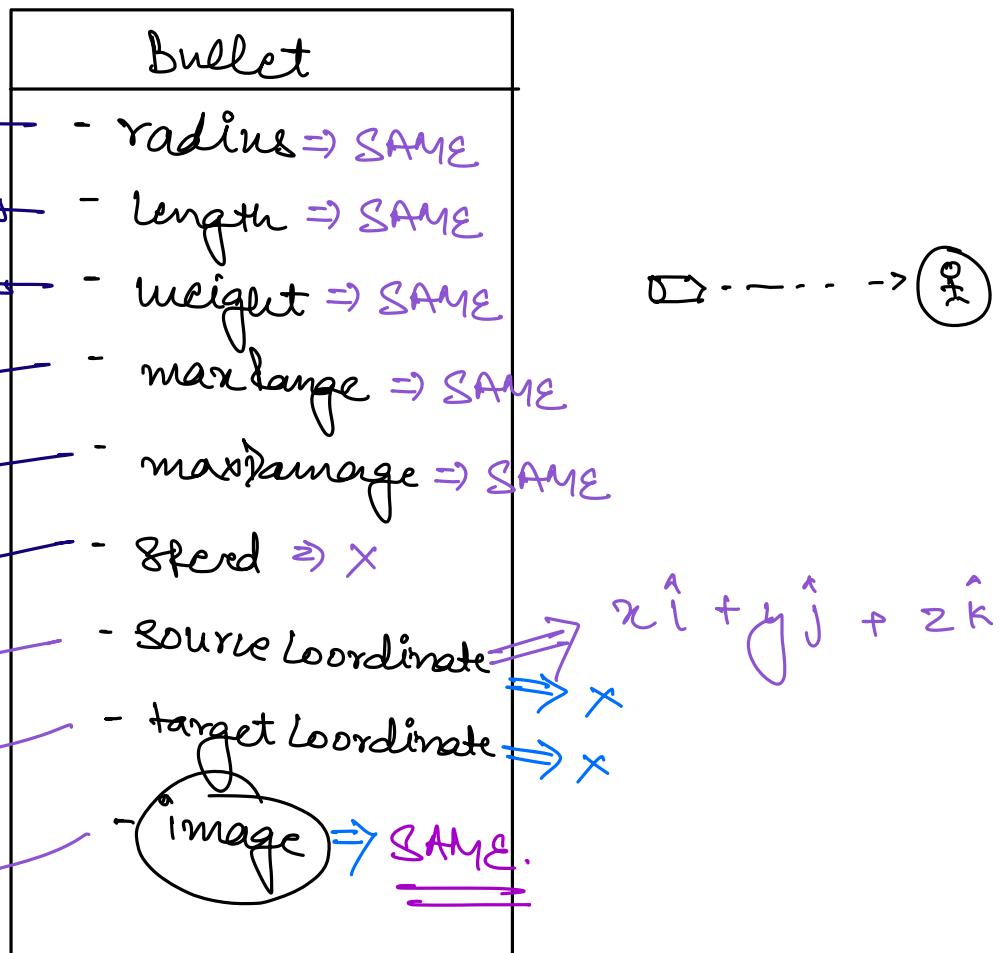


⇒ At the start, game loads the data for each player.

⇒ Complete state of the game is downloaded at start time & after that all the events will be transferred to other players.

$100 \text{ Player} \times 10^3 \text{ Bullets.}$

⇒ 100,000 bullets.



$$\text{Size of 1 Bullet obj} = 9GB + \underline{\underline{10KB}}$$

$$\approx \underline{\underline{10KB}}$$

$$100,000 \text{ Bullets} \approx 10KB \times 10^5$$

$$(10 \times 10^5) \text{ KB}$$

$$10 \times 10^2 \text{ MB.}$$

1 GB.

⇒ To type of Bullets.

1 type ⇒ 10,000
=====

⇒ Bullet Objects have lot of common attribute.

⇒ There are 2 type of properties.

Intrinsic
=====

Extrinsic.
=====

⇒ Same for all
the object

⇒ Not same for
all the object.

Flyweight-
=====

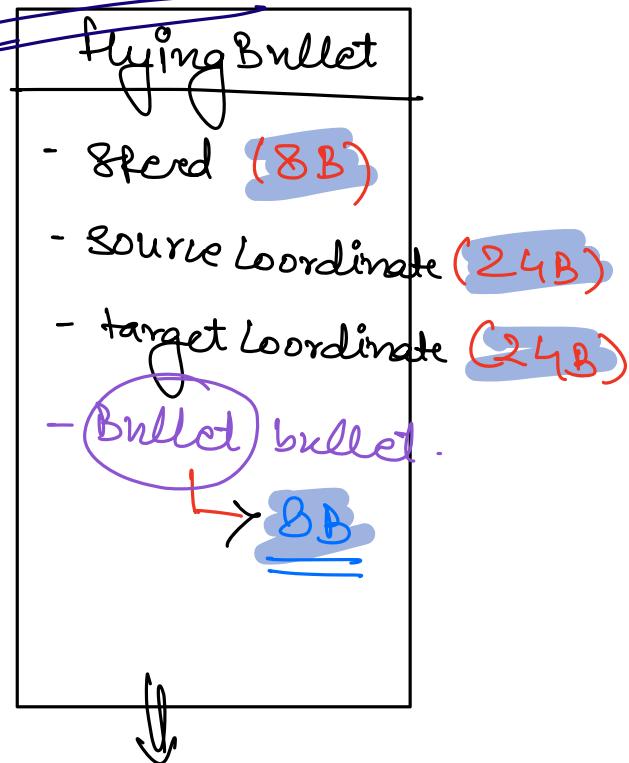
If we have a class whose objects contains both intrinsic & extrinsic properties & lot of memory is being consumed because of this, consider using Flyweight.

⇒ Divide the class into ②

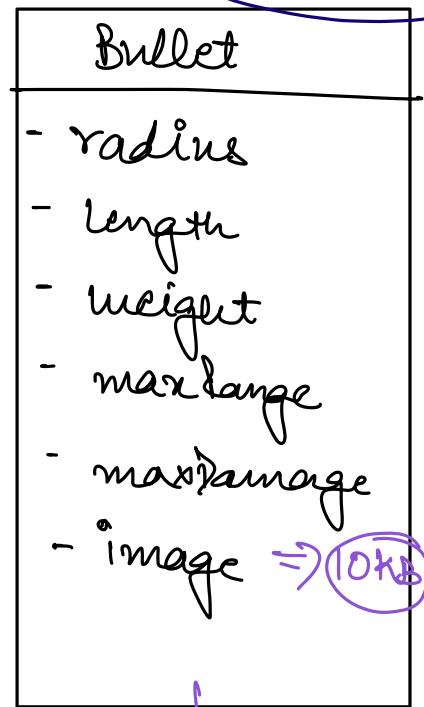
→ Class with only intrinsic

→ Class with only extrinsic.
=====

Extrinsic -



Intrinsic



10^5 Bullets

1 Object

$$\text{Size of 1 Obj} = \underline{\underline{64B}}$$

$$\text{Size} = \underline{\underline{10KB}}$$

$$\text{total size} = 64B \times \underline{\underline{10^5}}$$

$$= 64 \times 10^2 \text{ KB.}$$

$$= \underline{\underline{6.4 \text{ MB.}}}$$

$$\text{Total space} = \underline{\underline{6.4 \text{ MB}}} + \underline{\underline{10 \text{ KB.}}}$$

7MB.