Smart Pointers

[auto memory management]

↓

garbage collector

→ Can we directly init a variable in heap in C++??

↳ No,

↳ We have to always use pointers

free / delete → they are working on ptrs.

1. Unique Pointer
2. Shared Pointer
3. Weak Pointer

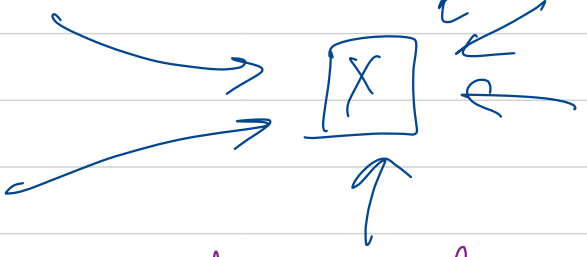
→ 3 types of Smart Pointers

if we have created an object using unique-ptr, then no other pointer can point to it.

Shared-pointer gives you access to shared objects

→ Destructor → Structs → this is a function which is called finally before the struct is removed from memory.

Smart ptrs → reference count → 0 → destroy it



weak ptr → doesn't maintain reference count

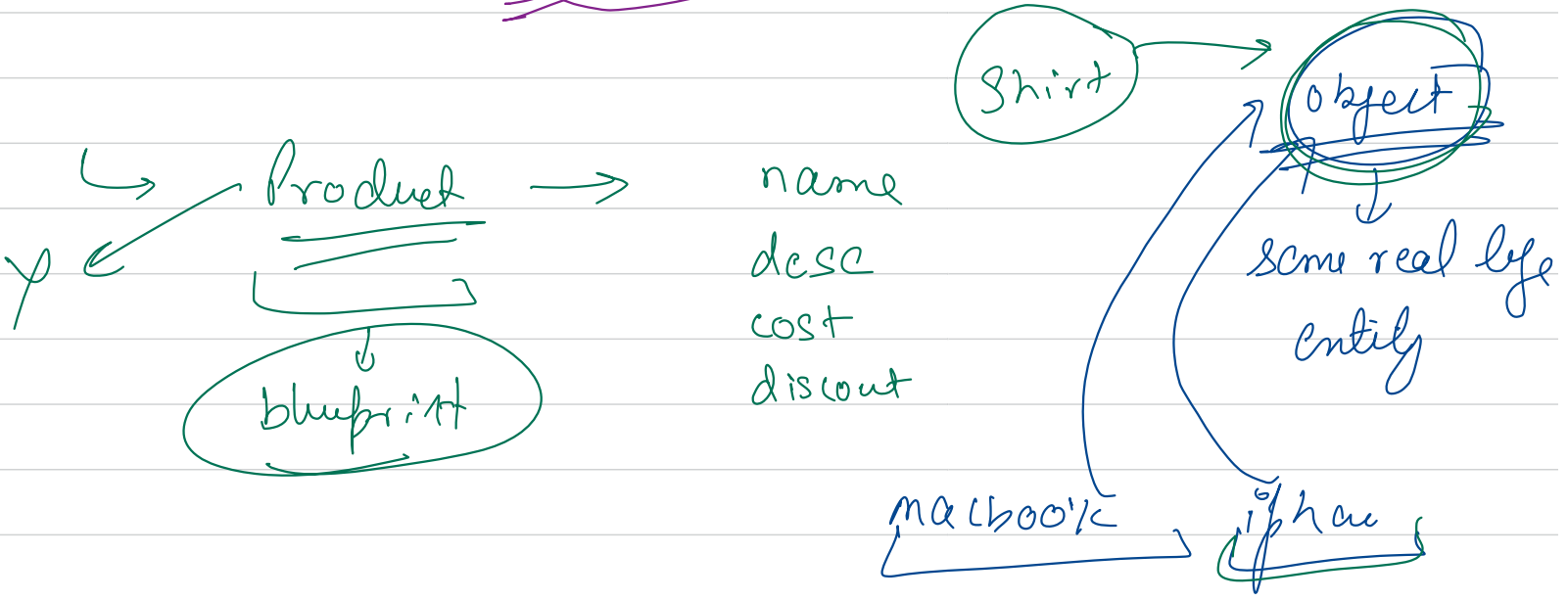
dangling ← weak ptr Solve the problem of a dayley ptr

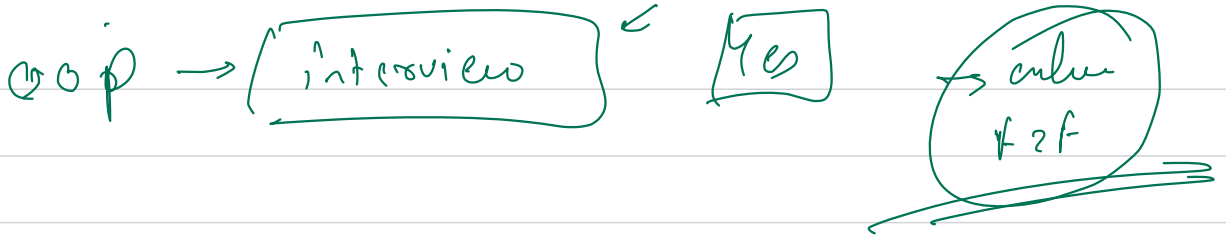
expired()

↳ any piece of code that is common for all products

features

OOPS




GDP
interview
Yes
culture fit

Objects → real life entities / also called as instances

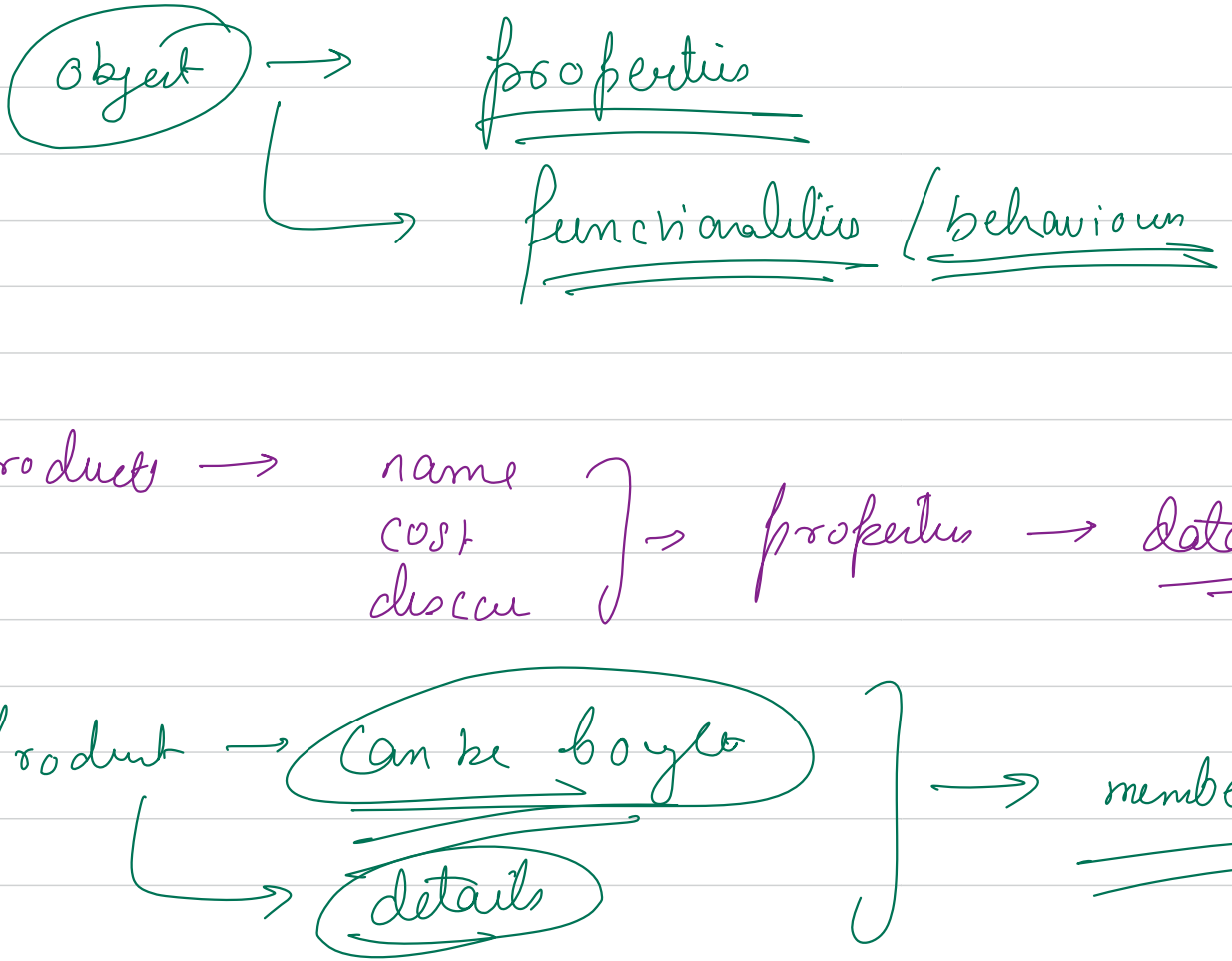
Class → is a blue print of a real life Objects

↳ a huge upperhand in only business logic

float f = 3.14;

→ fp number data type

Cardekho.com → vehicle → cost → type float


object
properties
functionalities / behaviours
Product
name
cost
discount
properties
data members
Product
can be buyed
details
member func^n

Constructor → it creates a new object instance of a class

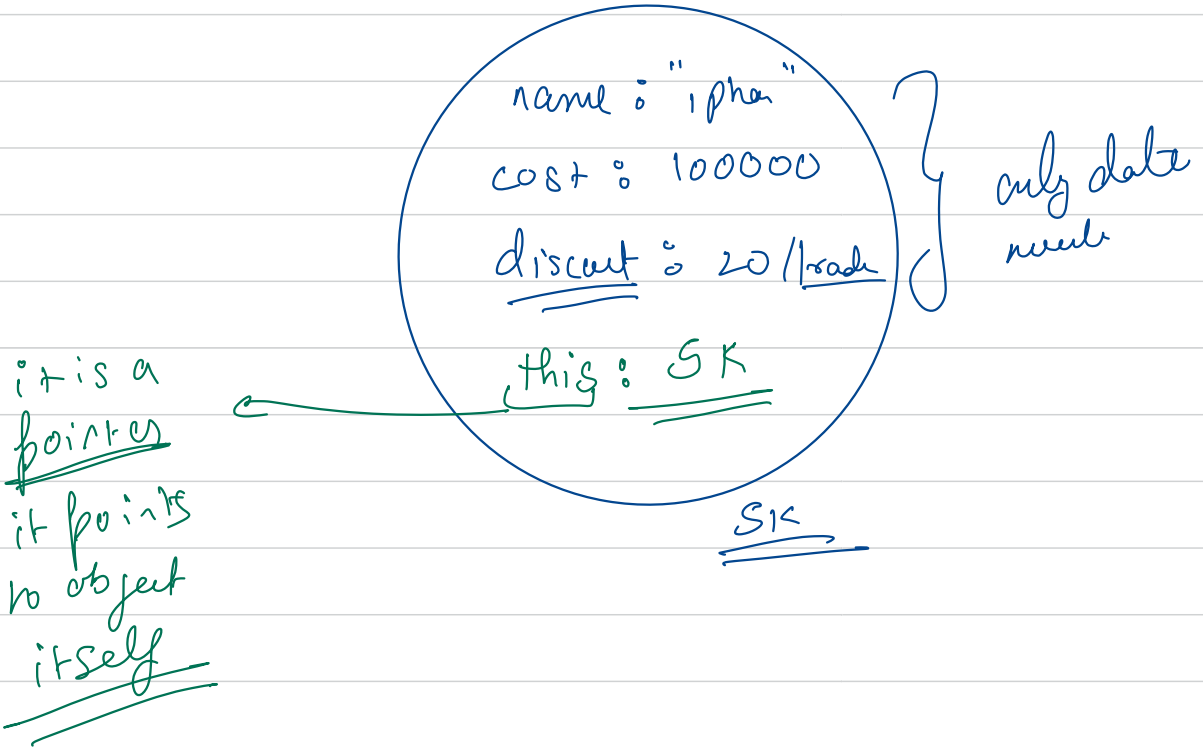
default Constructor

visibility

1. private → default
2. public
3. protected

→ inheritance

what happens when we create an object ??



if accessing via a pointer (→)

ptr → name

if accessing by object (.)

Obj.name

only diff between struct & classes is that everything in struct is public by default but we can have custom visibility modifier in classes.

We can have private & protected in structs also