OOPS Class *

Classes & Object

Constructor & Deconstructor

Static, this

Relationship

Super

Polymorphism

final

10:30 sleep *
wake up at 5 *

today's class

new → memory allocation

1. Space
2. Worker (Map) — step-by-step
3. Changes

Value initialization

STATIC * { memory Management } — Part of MAP

*** { Calling without object Creation }

MAIN static, so that we could (or do not have to initialize space for everything else!

like AA, LL 2 etc!

Class

*** method
variables
class

Egg *

***

Run → Input
↓
String [ ]
↓
main (String [ ])
↑
dynamic location

things *

imp for coding music

① Map → fabli static bnn gya.

② Object

non static → only after object creation

space already assigned

STATIC → STATIC

STATIC → NON-STATIC

{But not to this guy}*

{#If you wanna call non-static from main}*

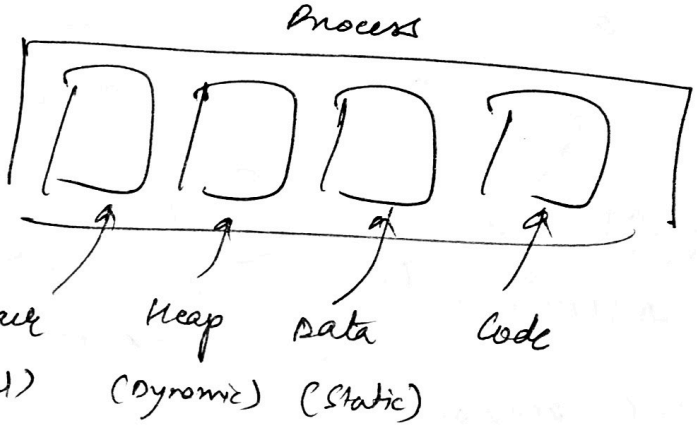
↳ Make an object of that class inside main class only & then use it.

NON STATIC → static

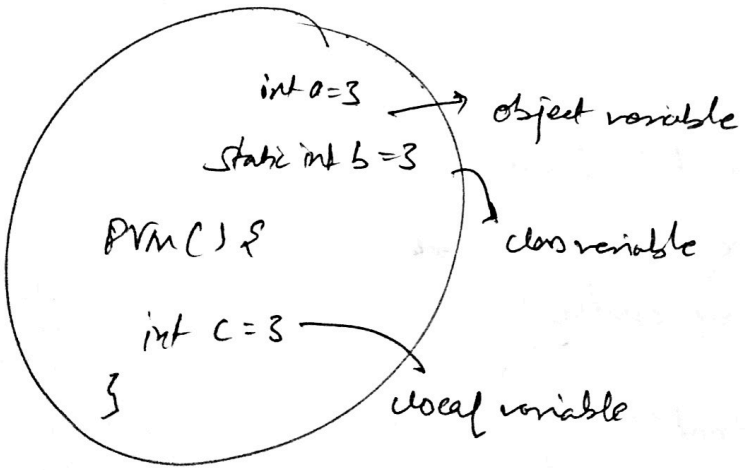
→ non-static

{JA Pym Java}**AA

OS



this keyword



this();

Constructor calling

↳ always first line in the method!

always value initialization

return after doing that stuff

Relationship

ASSOCIATION → object creation

INHERITANCE (Math is a trigonometry)

AGGREGATION (Weak) (Math has a trigonometry)

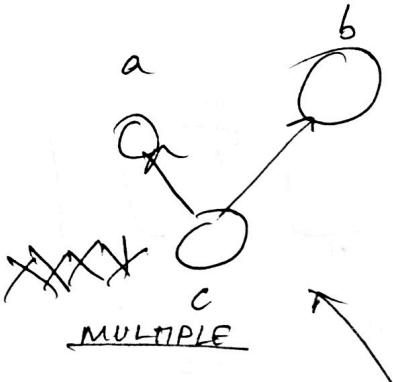
COMPOSITION (Strong)

car {
  engine() → Strong
  audio() → weak
}

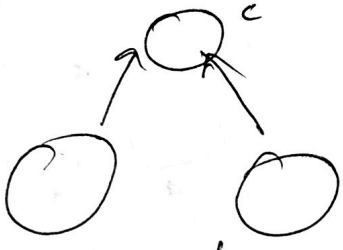
Types *

SINGLE

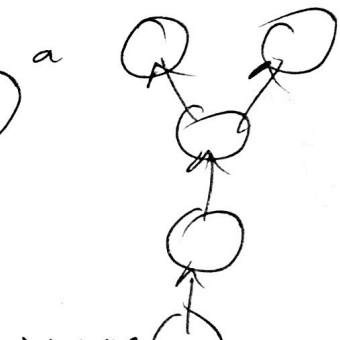
MULTILEVEL



MULTIPLE



HEIRARICHAL



HYBRID (combination) *

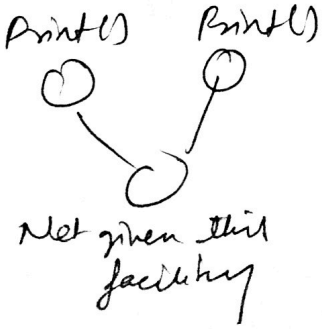
#Super() keyword?

extend ke time pr comparator ke andar

Parent class ko init krne ke liye ye current class ke andar hota hai!!!

Why not in Java??

More code & complexity! **



Not given this facility

Polymorphism

many ↙ ↘ behaviour

ONE method

↓

multiple behaviours

↙ ↘

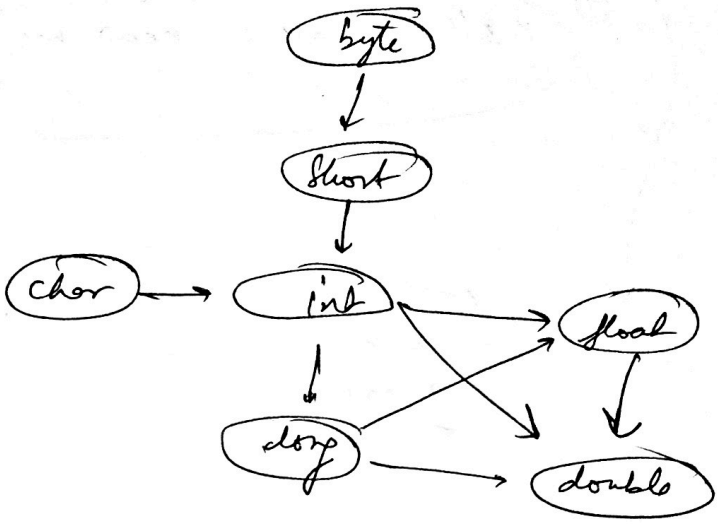
Overloading (Static Compile) binding

- change signature
- same class
- Resolution on the basis of typecasting

Overriding (Run Dynamic) binding

- Different Class
- Resolved during run-time!

Typecasting



(Pretty cool) ★

@Override

guarantees that similar method exist in parent class! ★★★

same signature!

MOST IMPORTANT

4 combinations

methods override ✓ (only) ★★

variable override ✗✗

```
msflat Obj1 = new msflat();
msflat Obj2 = new msflat();
```

PROOF

ONE DIRECTIONAL GAME HAI YE TOH

LHS → variable — compile time

RS → methods — run time

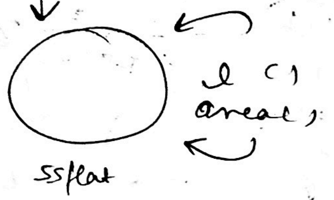
different variable output
similar method output

③ ssflat obj3 = new ssflat();

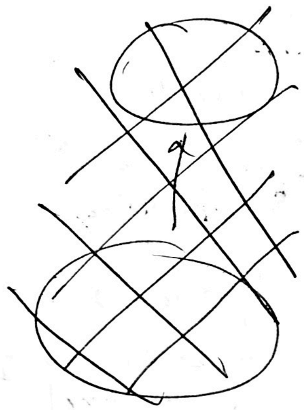
④ msflat obj4 = new msflat();

we do not have object for msflat
Hence, no access!!

[error] according to me!

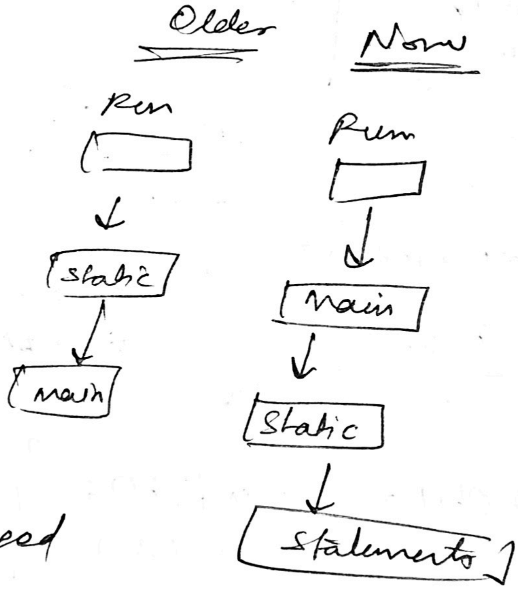


(msflat kabhi bna★
hi ni isme.



STATIC BLOCK ★★★

```
static {
    sysout ("Hello");
}
```



final

↳ variable → can't be changed
↳ method → can't be overriden
↳ class → Not inherited
↳ constructor XX

Make them stronger! ★★★

final static → static block for instantiation

~~Static~~
final int → constructor to instant

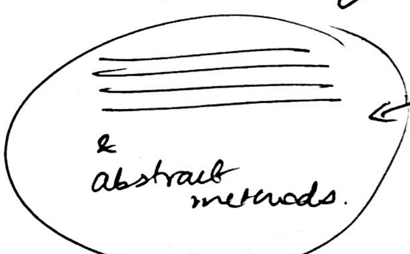
Data hiding Vs method hiding

No idea about
previous
Classes while
reusing variable
& methods in
Inheritance

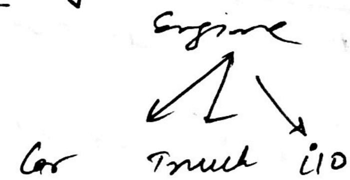
ABSTRACT

ss flat → lift nahi pta!

Abstract Class ? (for common functionality)



← constructors
possible
(since we have
variables)
& stuff




```
abstract ss flat {
    ≡ different methods & variables
  abstract lift ();

}
```

abstract/non abstract msflat extends ssflat

{ (1) → implement all abstract functions

→ or be share implementation of abstracts function. and not for some (be abstract class for life)

}

Interfaces in java

(100% Abstraction)*

Abstract class → functionality
→ hidden
→ Abstract

```
interface PQR {
    public void lift(int floor);
    public int area() {
        return a x b
    }
}
```

~~public int area() { return a x b }~~ ← not possible

ko'ice 10 Question

Q
why static??
{public abstract static} ***

interface is class of blue print!

{ interface → class → object } ***

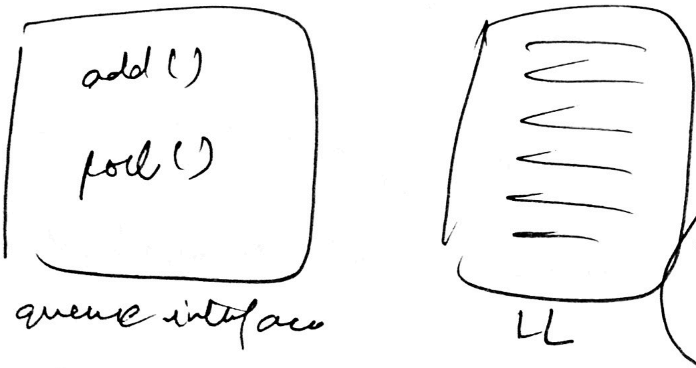
public abstract static **	lift(int floor);
public static final	int p = 3;

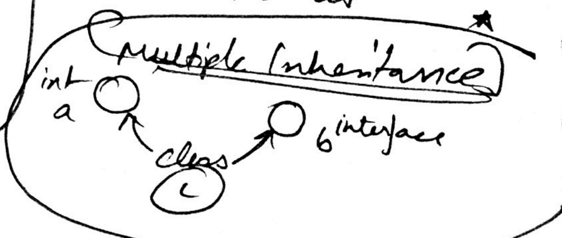
* pre-written

(#) Pretty cool concepts

✓ OOPS done!!

implement
class → Interface
Interf → Interface
extends





in java

** # way of abstraction

myqueue que = new LinkedList();

{ only two function implemented } **