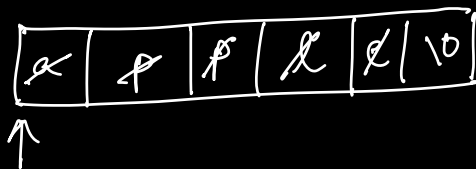1. String doubt & string performance
2. OOP basics — classes, objects, ——, this, static, final.
3. inheritance and polymorphism
4. ~~enums~~ Abstract classes & interfaces
5. enums
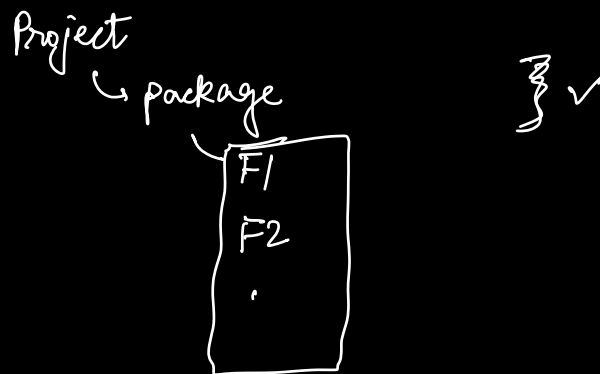6. generics.

• String   —   password

char []   —

1. String ⟶ encrypted string
   ⟶ immutable

   (s1) + "x" ;

   ⌊___⌋ memory

| x | p | p | l | l | \0 |

↑

2. printing logs — memory

   str
   Syso (str)

   Syso (array)
   ↑

private — (class) open to class

public — open to all   class Stud { ✓

default / — open to package private ___

friendly

Project
↳ package   } ✓

┌─────┐
│ F1 │
│ F2 │
│ . │
└─────┘

Constructor

Stud   s = new (Stud (—));

↓

Java

Java
(— name ✓ null
  — age   — 0
  — rollno — 0)

Stud   s   =   new Stud ("M", 1, 23);

Stack

print	this
print	this = 4K
main	S = 4K S. print ()

Stack

Heap

M
20

S = 4K

N
80

S1

O
40

S2

s
s1
s2

Static

class name

blueprint

class

data

class

O1

Objects
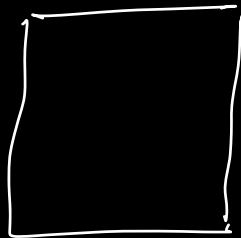
O2

O3
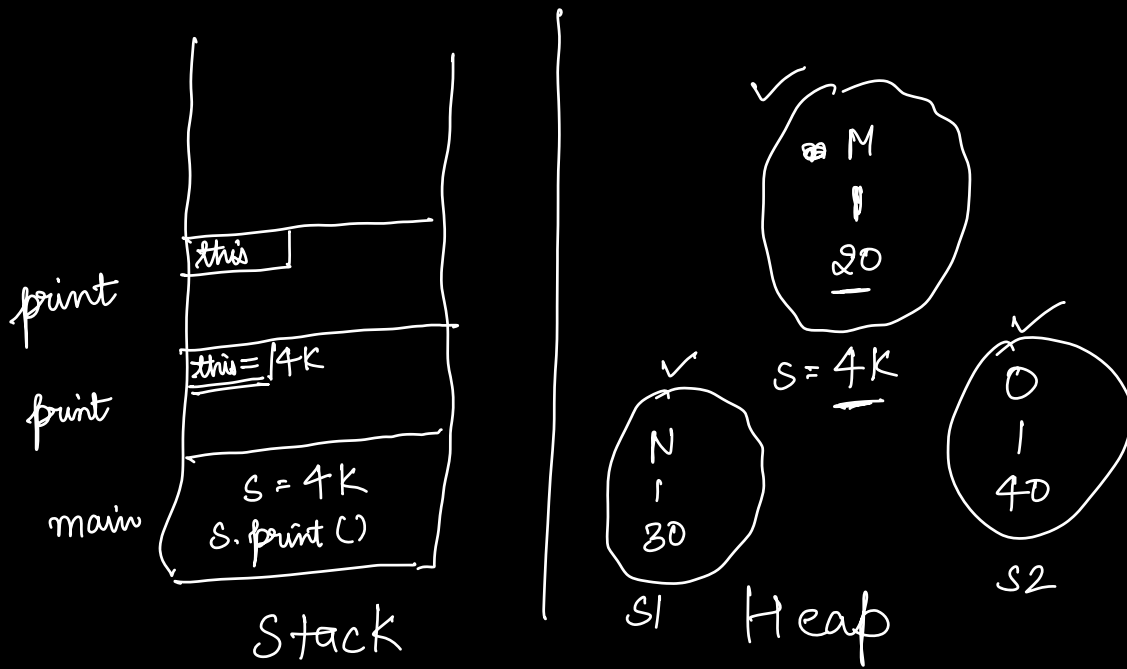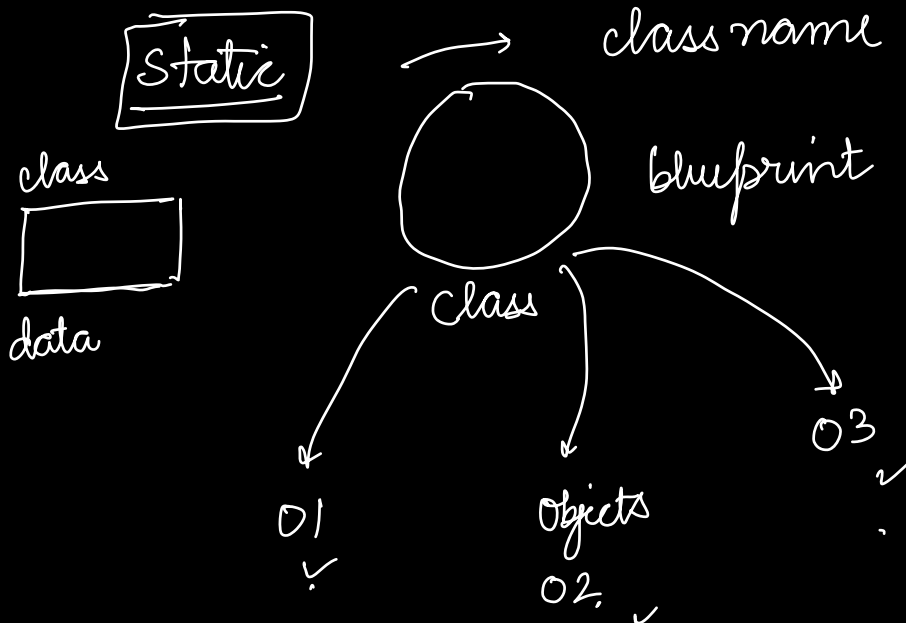
Static function → Static variables

→ Non static variables

<u>class name</u>

public <u>static</u> getcount ( ) {

<u>name</u>

}

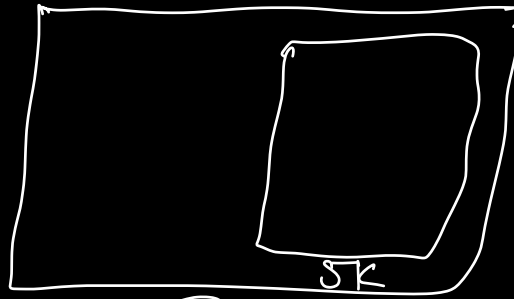Stud . getCount ()

Car ⟶ Vehicle

[ ]

Vehicle Ⓥ = new Vehicle ();

Car c = new Car ();

c = 4k

4K

4K

max
Sp

maxspeed

5K
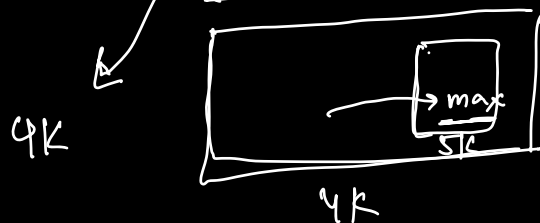
4K   sub

super = 5K

super ← in child class

this

→ public Car() {

→ super();

[   ]

default
constructor

}

Vehicle ✓   print ()

Car ✓   (Print ()) ✓

(BMW) ⟶ :

Vehicle ˅ =   new BMW () ;   ✓ //allowed

⊕ ⟶ String
   ⟶ Int

compile
time   [add (int a, int b)
        add (float b, float c)]   overloading

class student_calculator {

int add (int a, int b) {
& _____
}

add (float a, float b) {
}
}

add (5, 4)

+ ⟶ o

overloading ⟶ CT

overriding

Vehicle ⟶ print

↓

Car ⟶ print ✓