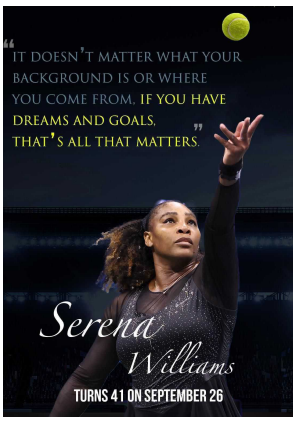
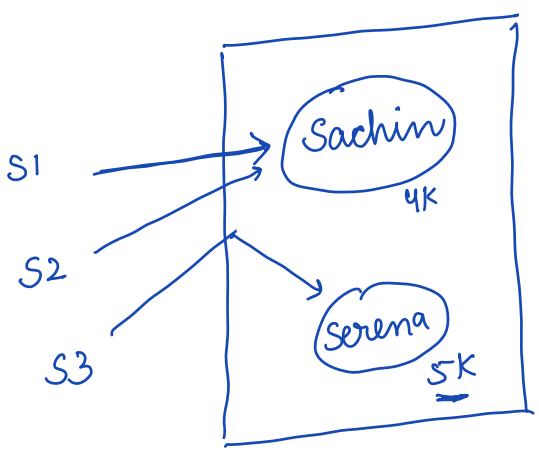


- Summary
- Today's content
- Strings

↓

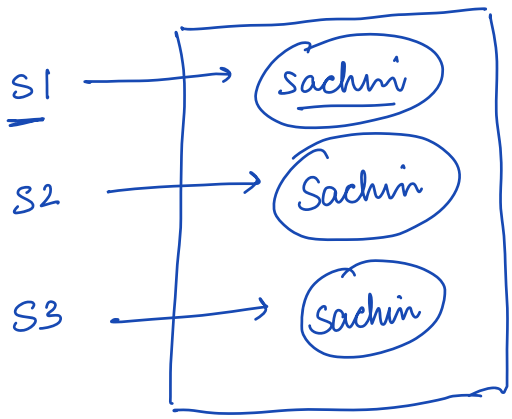
unchangable

if changed it would be new location.

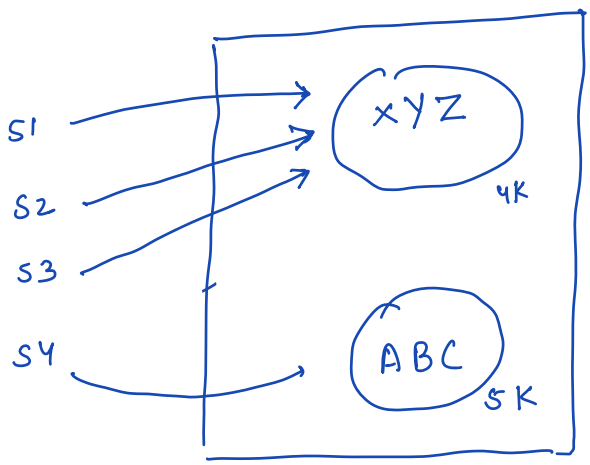


S3 = Serena

Case :

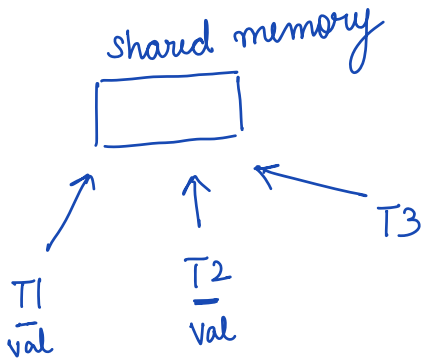


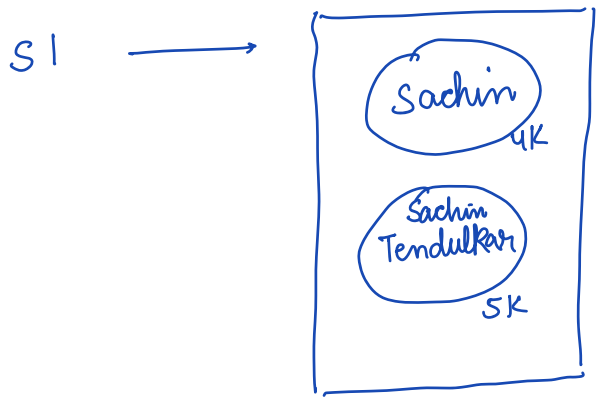
1. Save memory —



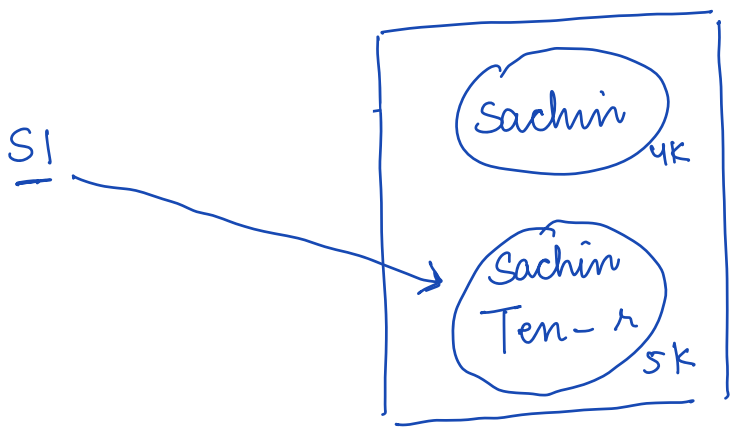
S4 = name = ABC

2. Thread safe





s1. concat ( )

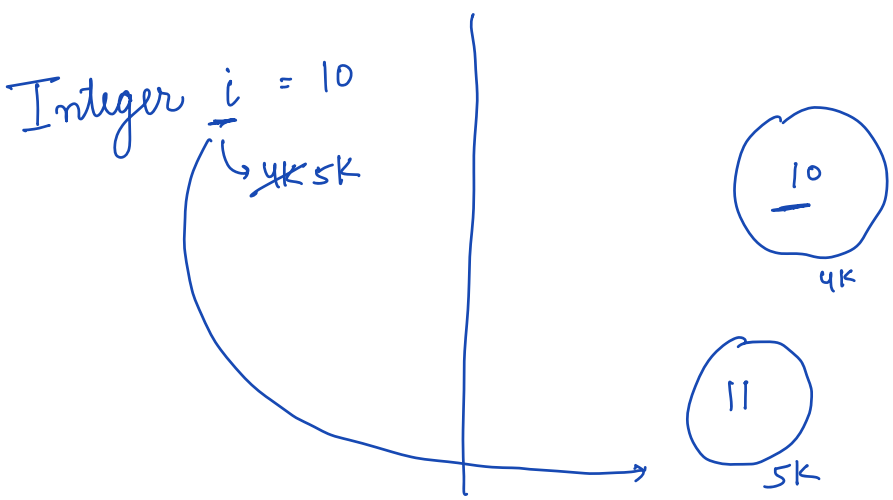


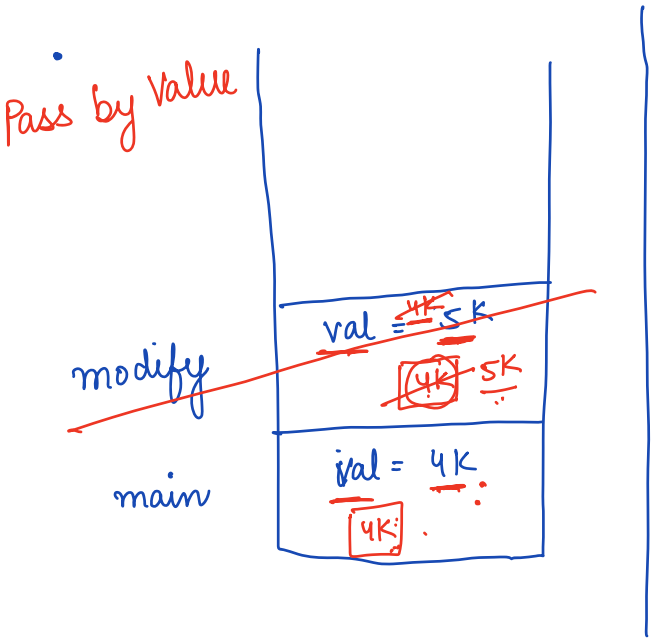
3. Class Loader → Classes → method Area → name

- Wrapper Classes
  - ↳ immutable in nature
  - Pass by Value
  - Class Integer {
    int i
  - class Character {
    char c

}

}





110
10

- Pass by Value — copies are created

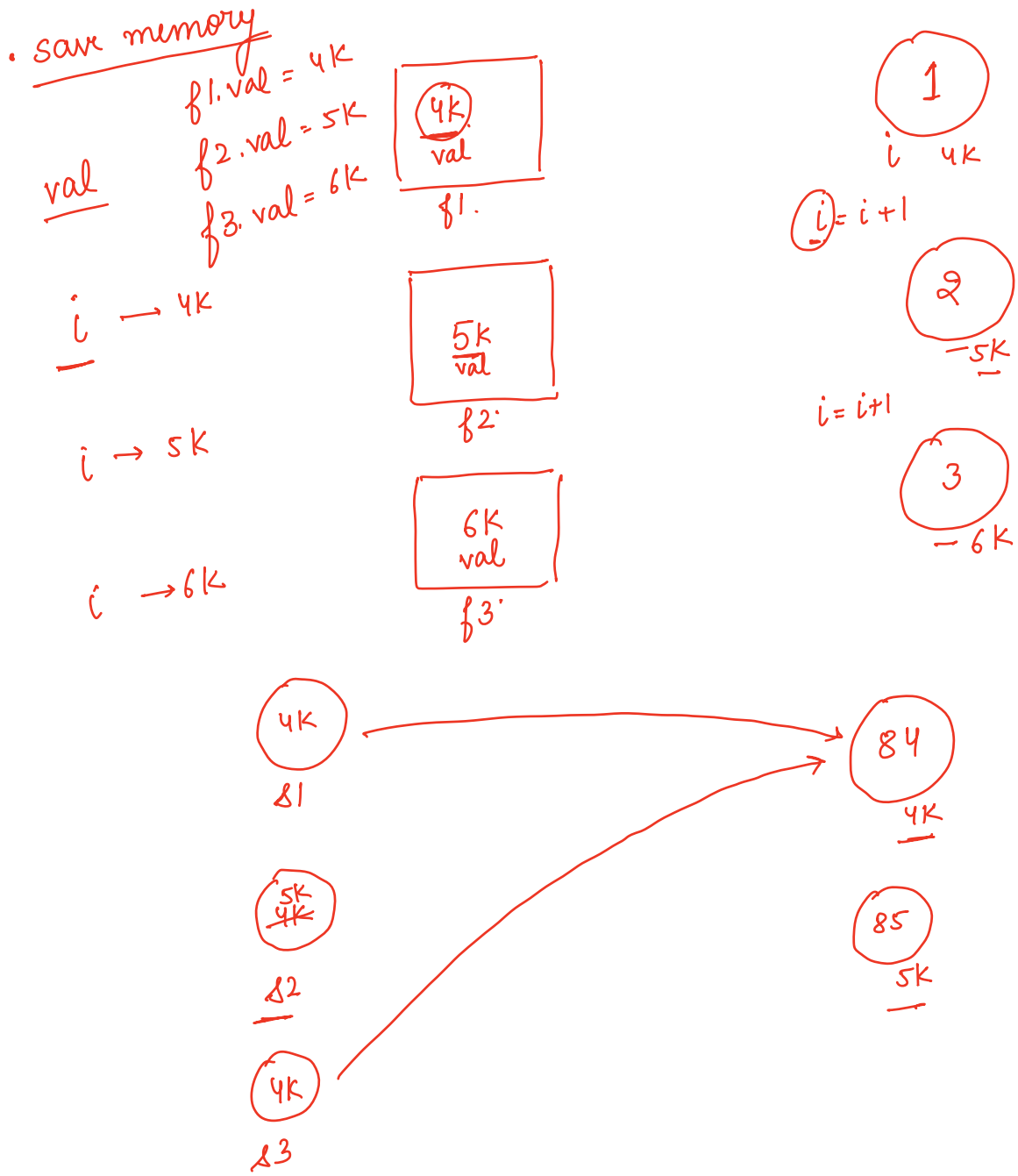
→ Object → mutable

4K → 10 → 11

4K

→ Wrapper Classes → Java

→ To Do: Find a method → custom classes


save memory
f1.val = 4K
f2.val = 5K
f3.val = 6K
val
i → 4K
i → 5K
i → 6K
4K
val
f1.
5K
val
f2.
6K
val
f3.
1
i 4K
i = i+1
2
5K
i = i+1
3
6K
4K
s1
5K
4K
s2
4K
s3
84
4K
85
5K

a Math question

. −40 % 9

maths = R = Divi-dent − Closest <= Divident

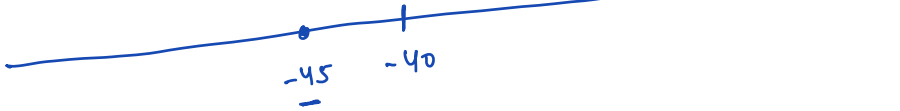
124 % 15 ⟶ 124 − 15×8 / 120

= 4

−40 % 9 = −40 − (closest mul of 9 <= −40)

= −40 − (−45)

= −40 + 45 = 5



−40 % 9 = Dividend − (Divisor × Q)

= −40 − (9 × −40/9)

−4.3...
9) −40

−4

= −40 − (9 × −4)

inbuilt −40 % 9

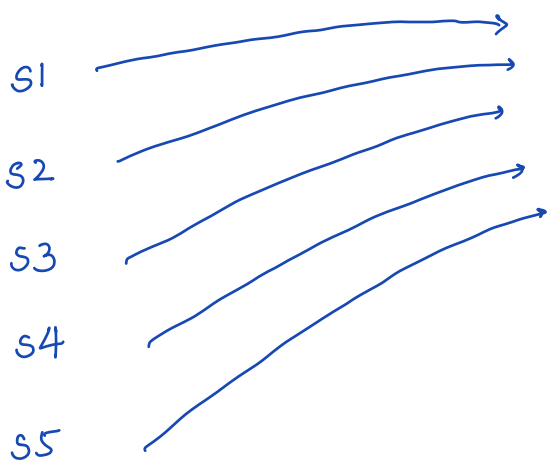
= −40 + 36 = −4

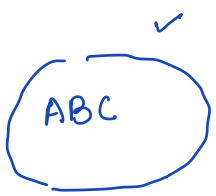
⇒ (x % m) + m

Static => class property

- variables
- methods
- blocks
- nested classes

common to all objects

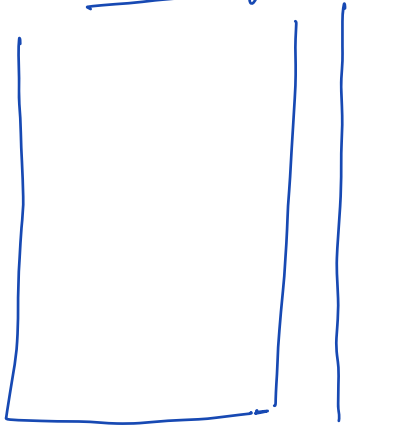


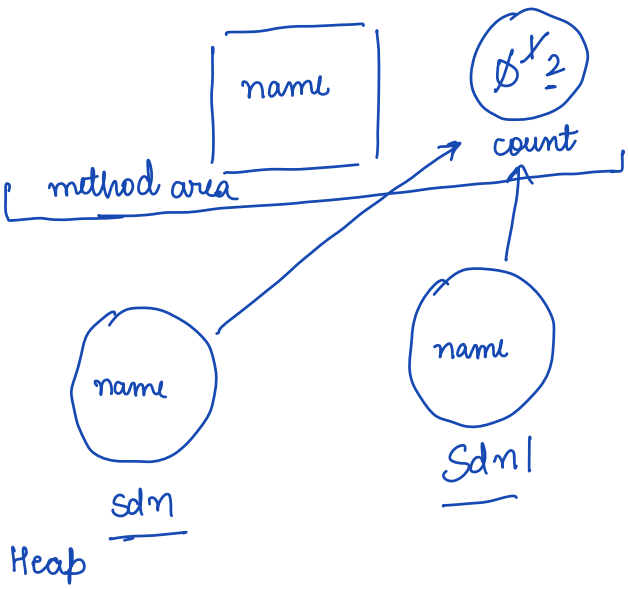


same value

visible to all

count of objects





Static functions

↳ common to the entire class

Math class — function

Math.pow(2, 4) → 2^4

Math.abs(-20) → (-20)

utility functions

print value of count of objects

↙ static ↘ non static

this - object reference

1) count of objects → no this

2) print Name → this ✓

main — ✓ static

- JVM — main — object independent
  ↳ classname.

Static function → Classname this ✗

↳ non-static. → ✓

↳ static variables ✓

Non static function

↳ Non static - ✓

↳ Static variables

break - 10:34 pm

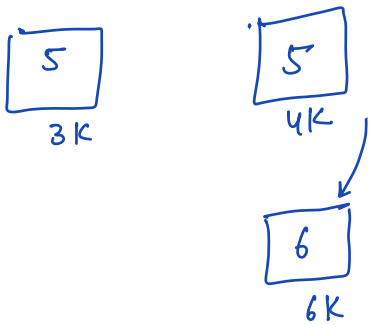
|

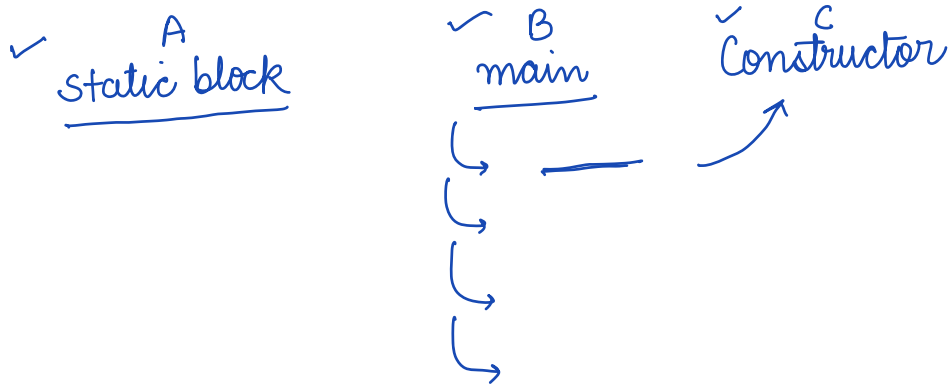
10:44 pm

✓ Give a video - HW

Q → ✓

Wrapper.





static {

To Do:

init block

INIT

}

- Static nested classes

class Student {

static class Ops {

variables

methods

}

static class → non static data

writing methods on static variables

Student.Ops.—

}

class Student {

String Name
int rollno

static count ✓
static String school ✓

f1 [Name, rollno

static class sfns {

f2

f3

}

}

class Straight line {

nested

static class Pair {

static functions f1
f2

}

}

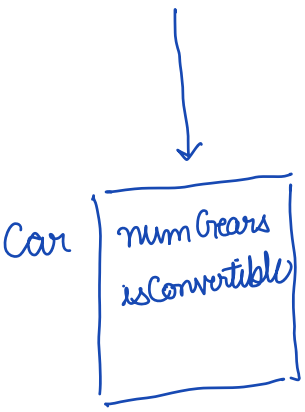
Inheritance — code re-use

Vehicle

Car

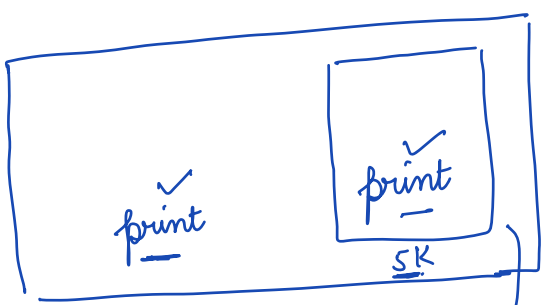
Vehicle.

colour
maxspeed
numTyres



Car c = new Car()

constructor
super()



c = 4K

super()

5K - object of parent

super.

parent class object

4K. getmaxspeed

4K. color

4K. print

super = 5K

private – same class

default – same package

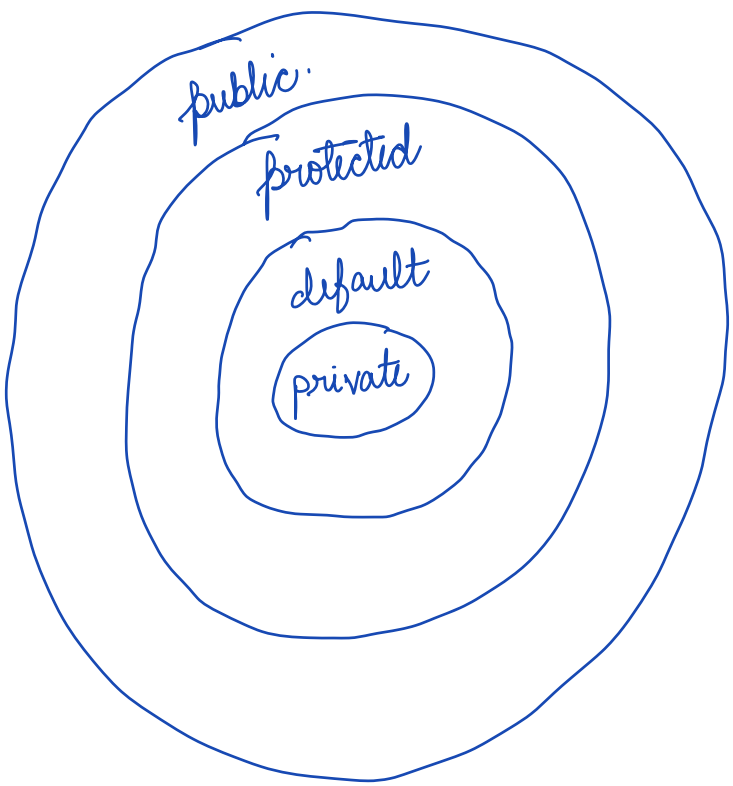
public – global access

protected – same package and

Truck
P1

Vehicle protected
P2

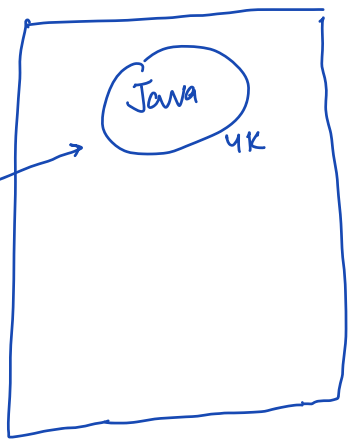
T e V



Doubts

String s1 = "Java". → 4K

String s2 = "Java"



s1 = "Javac"

checking → overhead

Pool

literal

new