$$\begin{tikzcd}
\text{OOPs} \arrow[r] \arrow[d] \arrow[dr] & \text{Programming} \\
\text{object} & \text{Oriented}
\end{tikzcd}$$

$\xrightarrow{}$ Classes & objects

<u>class</u> → blueprint / collections of objects

e.g. map → paper

object → Real entity / touchable → Building

Person → Male + Female

Rohan → object of person class

↓

- properties → Name, gender, age, Blood, Height/weight

Dog

- ~~bhevior~~ behavior

↓

Gestures → Walk/ talk/eat

Class Person {

└→ object →

↓

Properties → variables → name, age, r.n.

└ behaviors → methods → getName()

}

OOPs concepts / 4 pillars :-

1. Inheritance → Parent - Child

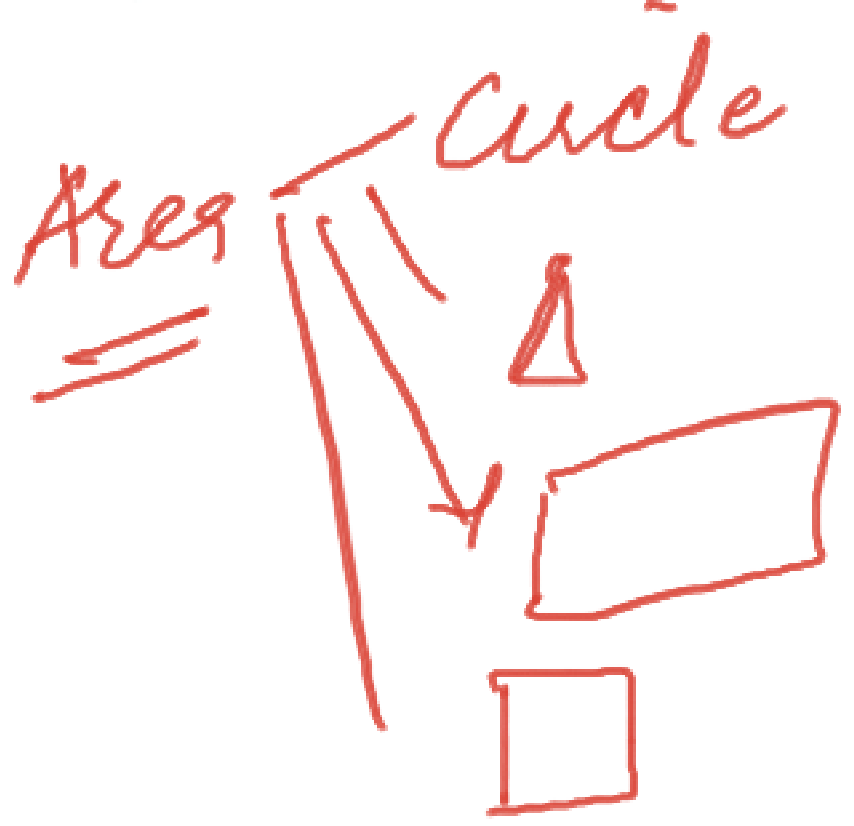
Calculator →
↓
Sc. cal.

Secure the Data

2. Encapsulation → Capsule → Collection of medicines

3. Abstraction → Hide the Implementation
↳

4. Polymorphism → many forms



Veg. → [Wash]

① Variables → container that holds the value

Syntax :- declare → int a;

define / initialise ⇒ a = 100;

main ( ) {

}

declaration along with initialization } int a = 60;

Types :- (1) instance variable →

(2) static variable

(3) Local variable

(4) Reference Variable.

I) Static → using static Keyword

non-static / — no use of static Keyword
instance

```
public static void main(String[] args){


}
```

Static Area/Method

$$\underbrace{\text{Non-static}}_{\text{instance}} \text{ variables} \xrightarrow{} \text{Static Area/Method}$$

can't access it directly, but can be accessible with the help of class object.

Class object:-

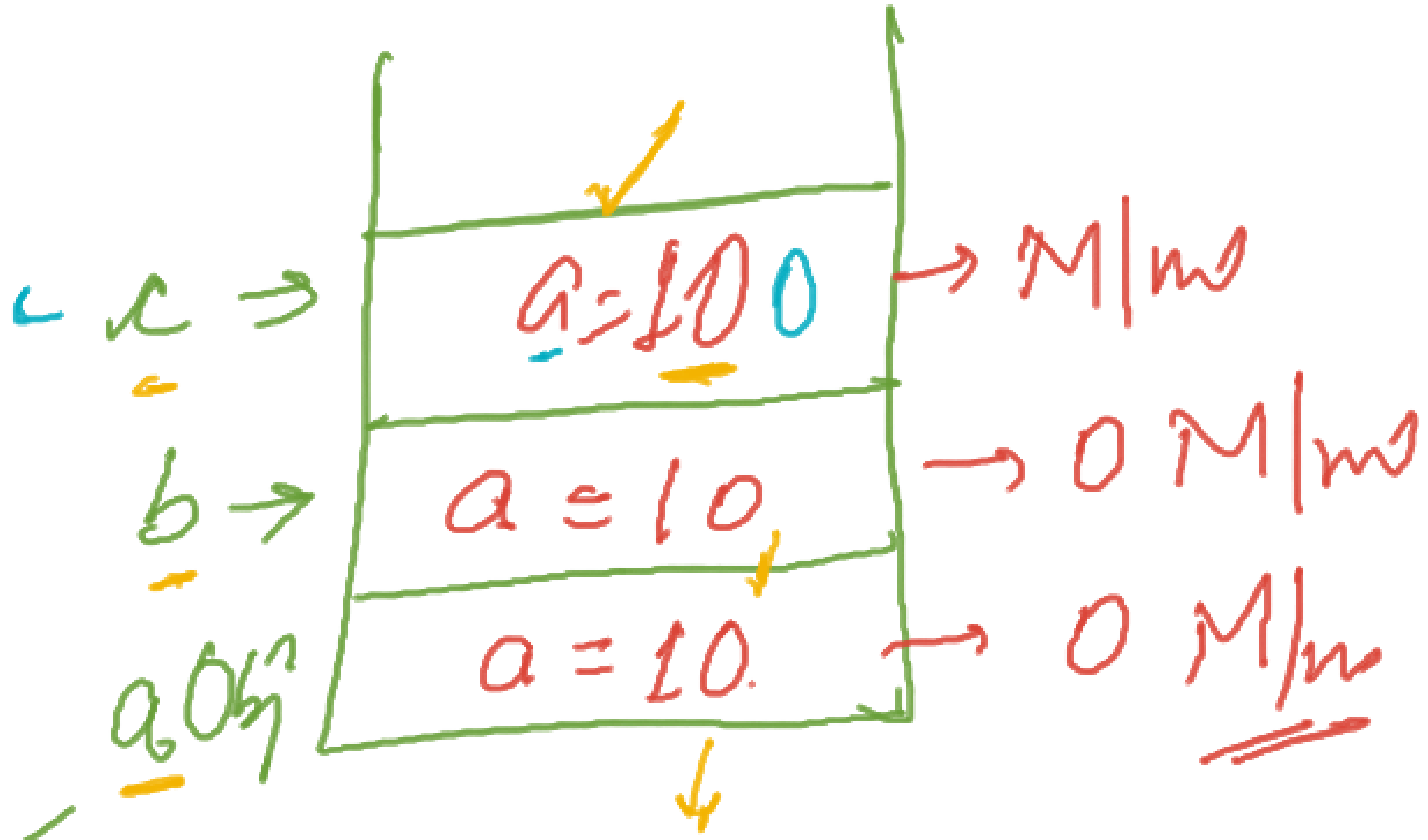
```
Scanner obj = new Scanner (System.in)
ClassName any-var-name = new className ( );
```

Value of instance variable varies as we create diff object of class.

A →

```
[ int a = 10;
```

instance /object level



c

```
ClassA aObj = new ClassA();
ClassA b = new ClassA();
```

c.a = 100;

Teacher → paper → Corrections

S1 → papers Corrections

S2 → papers Corrections

S3 → papers Corrections

S4 → papers Corrections

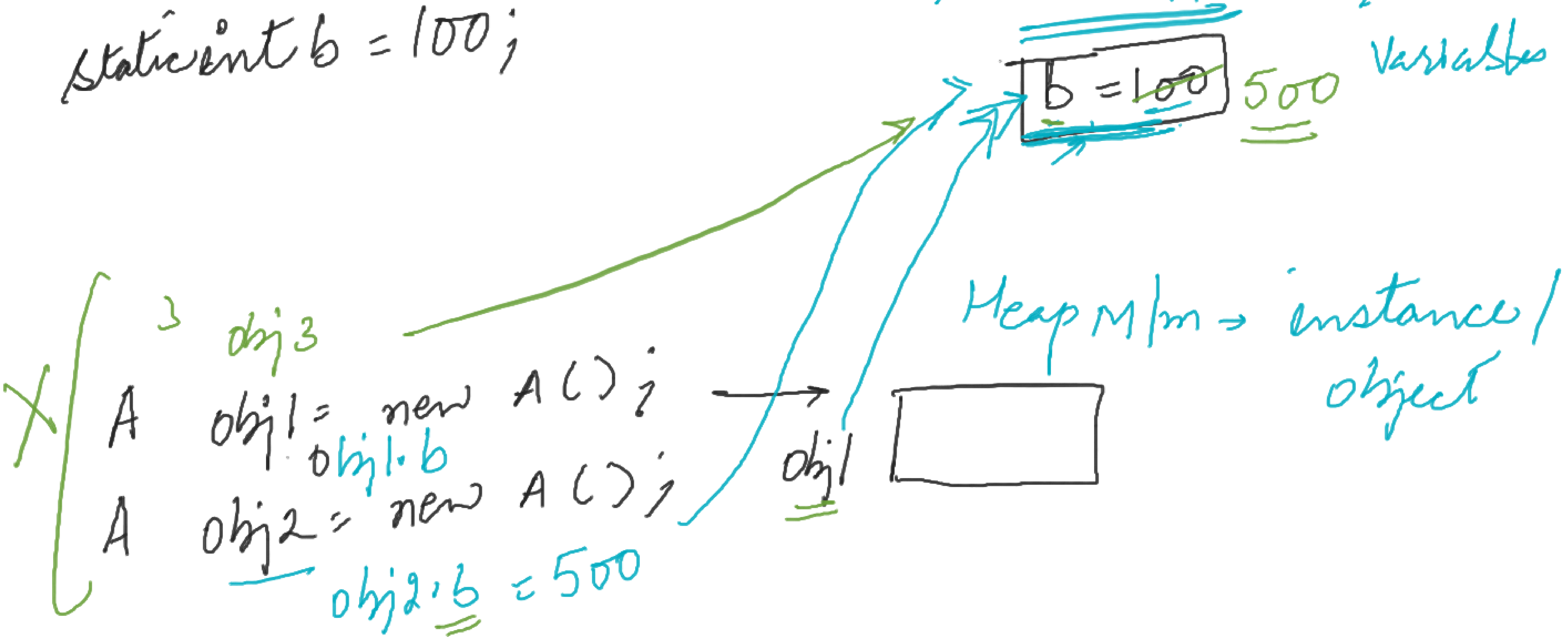
S5 → papers Corrections

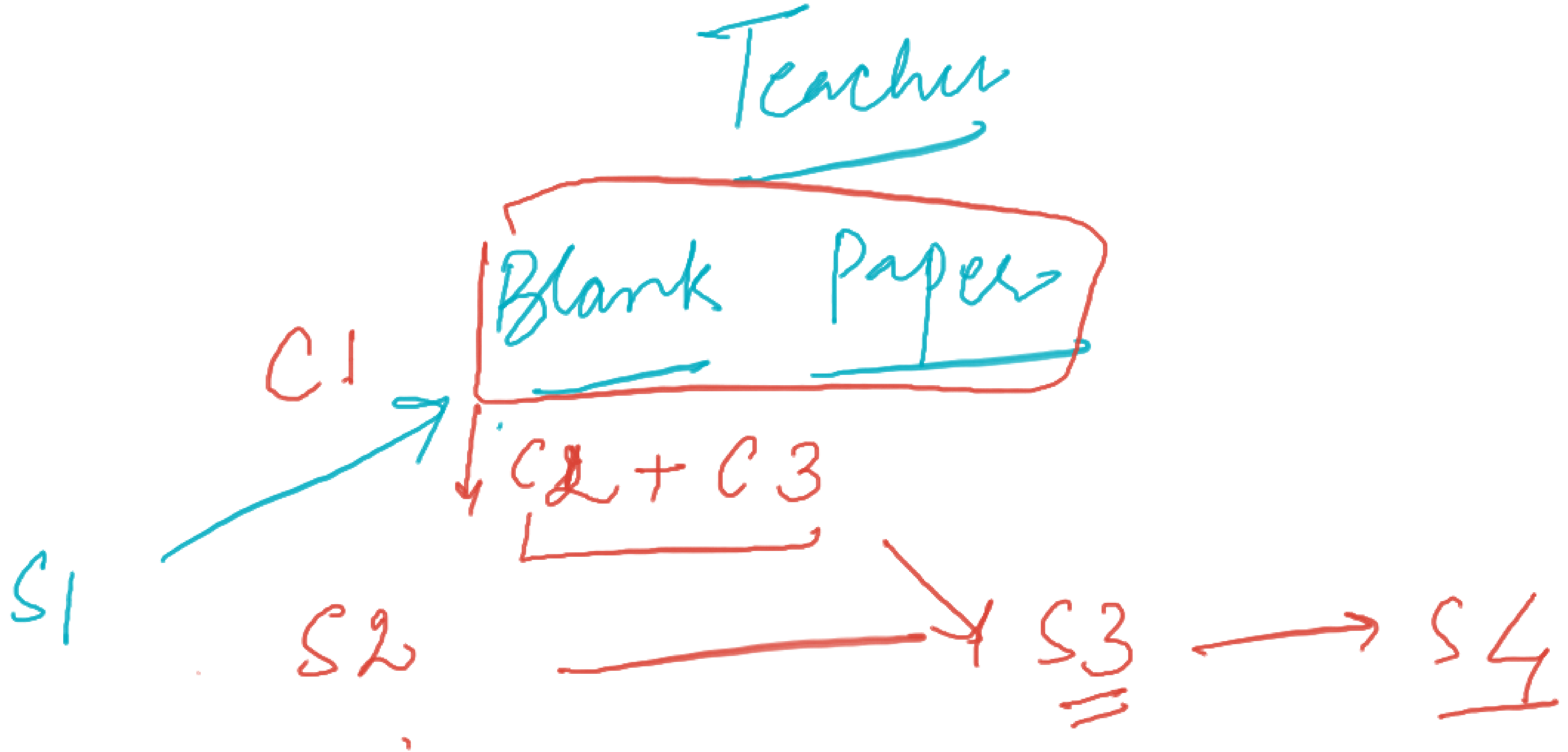
Static Members/variables Share the Memory.

How? class A {

static int b = 100;

}




Teacher
Blank Paper
C1
C2 + C3
S1
S2
S3
S4

Static M|m gets initalised / M|m allocation at the time of class Loading.

→ Source Code → A.java

→ Compile → javac A.java → Byte Code (.class)

→ Run → <u>JVM</u>

$$\begin{tikzcd}
& \text{A .Class} \arrow[r] \arrow[d] & \text{Byte Code} \\
\text{psvm} & \text{JVM} \arrow[l] \arrow[r] & \text{Class Loader}
\end{tikzcd}$$

static ← psvm → main

always start the execution from main()

(Loads the class

↓

(1) Static → Static M|m

(2) N.S → Heap M|m

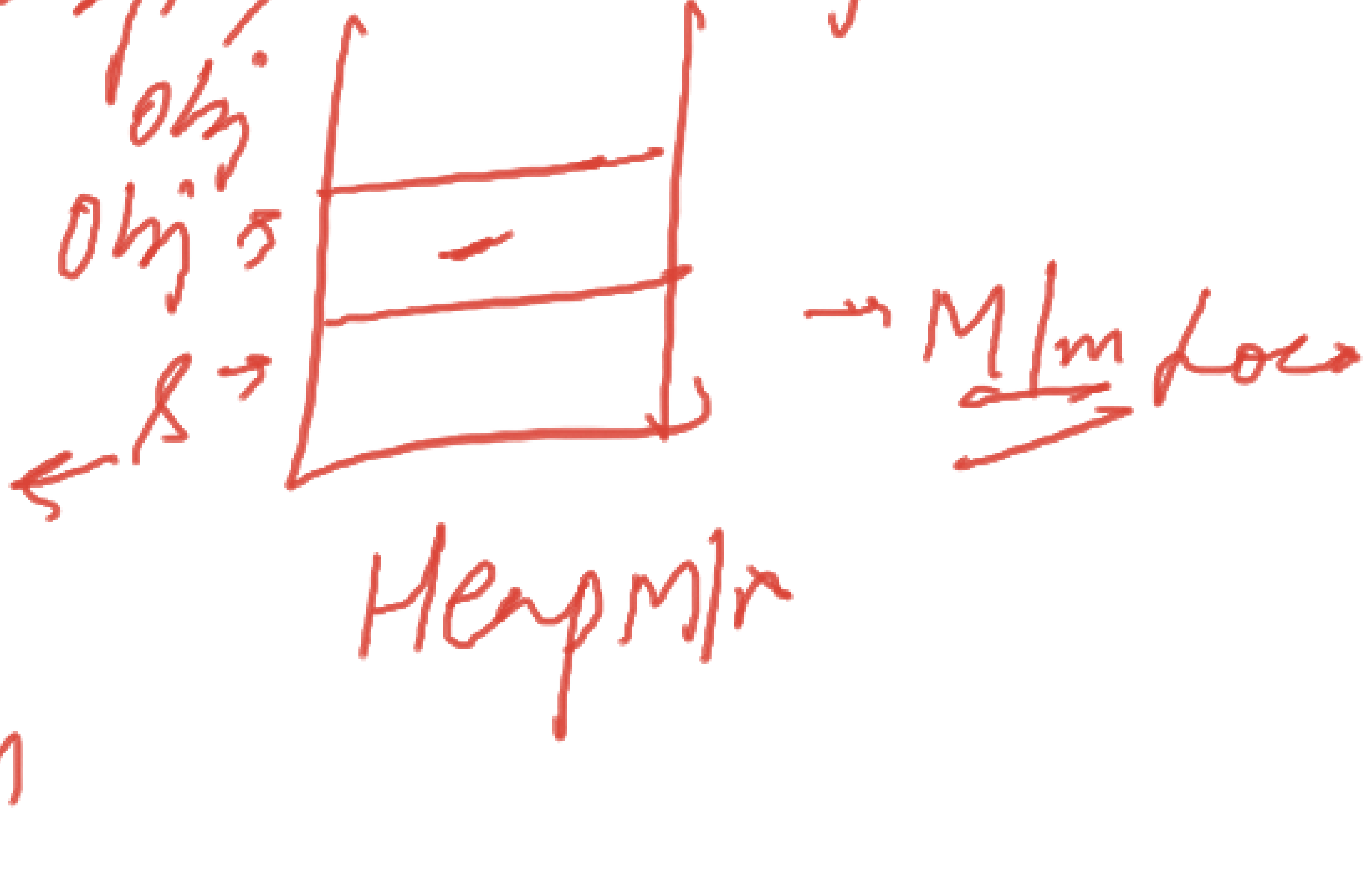
(3) Local → Stack M|m

```
# String s = new String ("Test");

Class A  obj = new class A();
```

Heap M/m → add

Class A obj; → unreferenced var.


obj
obj =
s →
Ref. M/m
→ M/m Loc.
Heap M/r