

parent class — m1() { }

child extends parent {

public void m1() {
 super.m1();
}

child C = new child();

C.m1();

// call the method
of child class

parent class &

parent (int &) {

}

}

class child extends parent

{

public child () {

super(100);

child c = new child ();

}

100

Constructor chaining this - refers the current class object

Class A Σ

$$A(\cdot) \in \Sigma$$

↓

3

$$A(\text{int } x)$$

this()

this()

3

Super \rightarrow parent class

→ ~~perfect~~
→ ~~complect~~

variable

method

→ P, C A

$a = \text{new } A(100);$

parent & child both having same value now

final keyword - restrict the user

↳

- to change value of var

- to extend the class

- method overriding

final
Static final

→ we declare ^{there and} init once only in ^{const} final -
→ init is mandatory and
can't change its value

finally → block | used in Exception
finalize() → despite of the Exception,
this block will be executed.

↓
finally
by default call by GC collector, to
free the M/m from unref. objects.

Modifiers:

- private → within same class
- default → within same pkg.
- protected → within same pkg + outside the pkg in child class
- public → anywhere

→ pkg 1

A →

B →

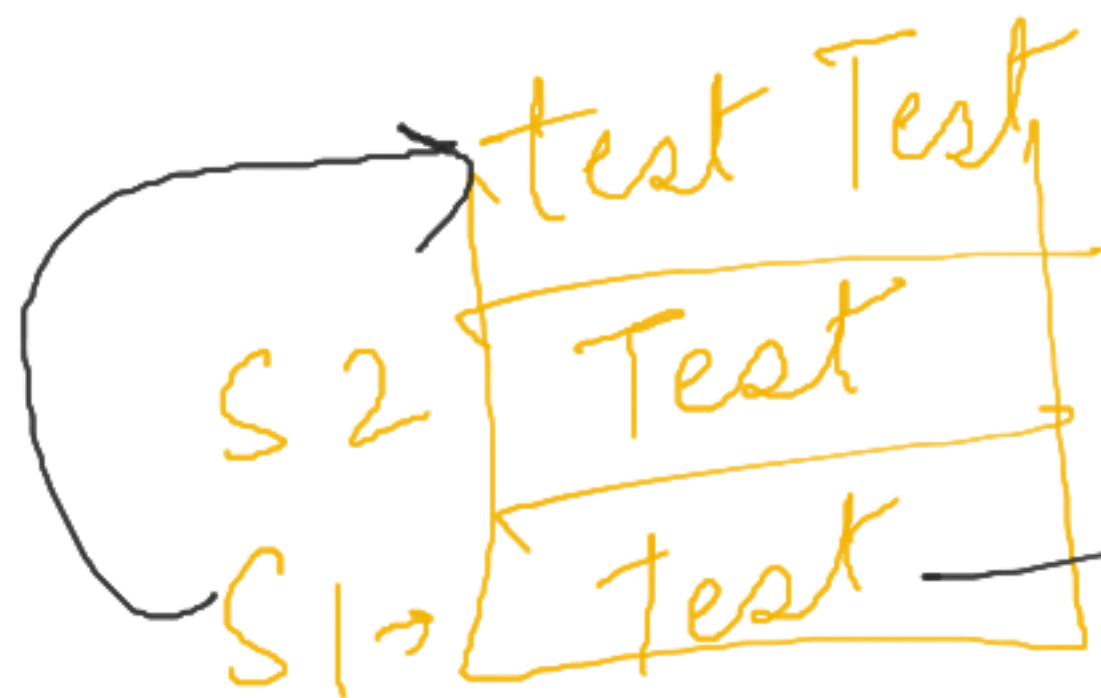
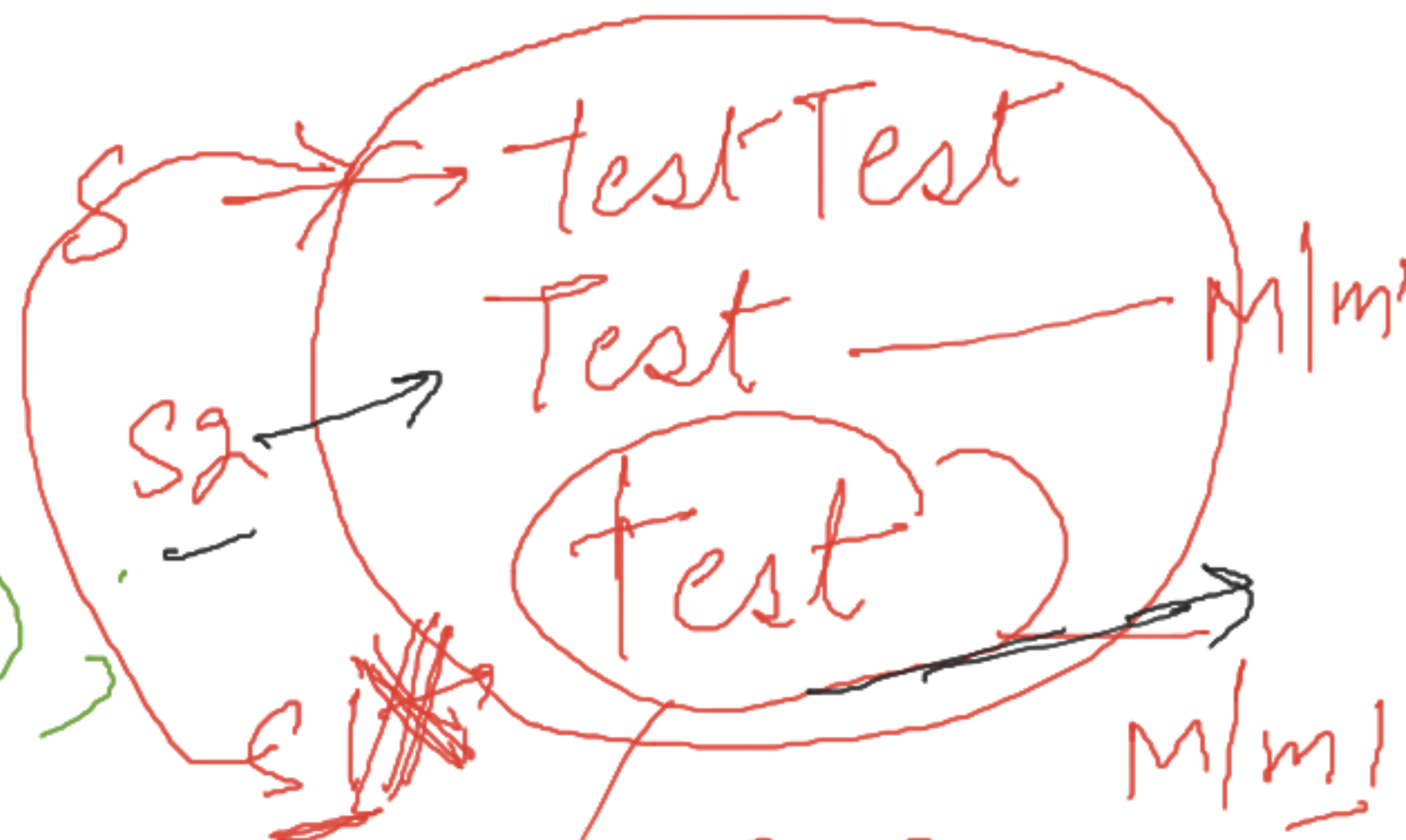
C extends A

pkg 2
D

E extends A

String S1 = "test";
" S2 = "Test";

S1 = S1.concat(S2);



Heap

garbage
unref

GC

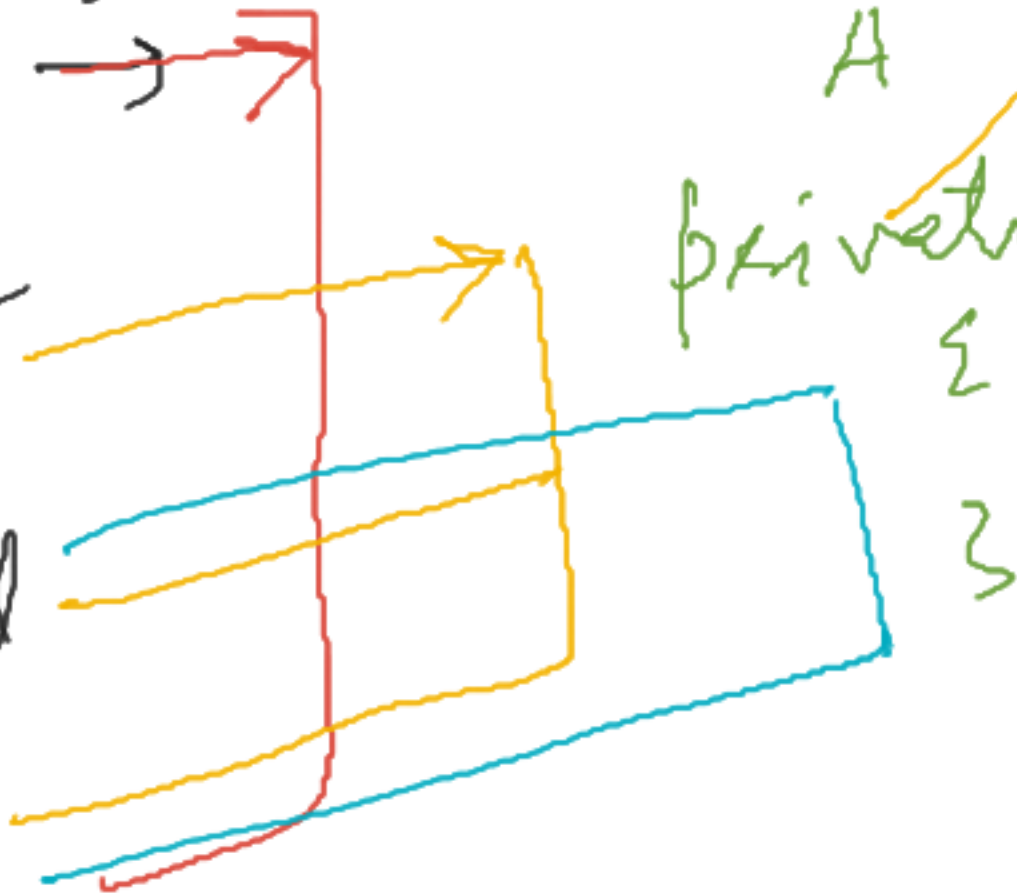
unrefs object

Can we reduce the visibility of public modifier to default?

Can't

parent class
private →

default
protected
public



child class

A protected
private void m1()

child
B public
void
m1()

private
X
Not allowed

Can we override main() ? — No

Can we overload main() ? — Yes

→ Static Members can't be overridden

→ main(String[] args) { }

Why main Method is static() ?

Why we pass String[] args ?

What is Run Time Arguments ? →

→ Static members are class level, no need to create an object
m/m allocated at time of class load

Why pass String [] args?

10, 20, 30

We can convert String into any other DT

By using wrapper class

parseInt(args[0]);

int A = Integer

How we pass runtime Args?

→ `java filename.java`

→ `java Classname 10, 20, 30`

Eclipse → Right Click → Run Configuration
args. ↓

