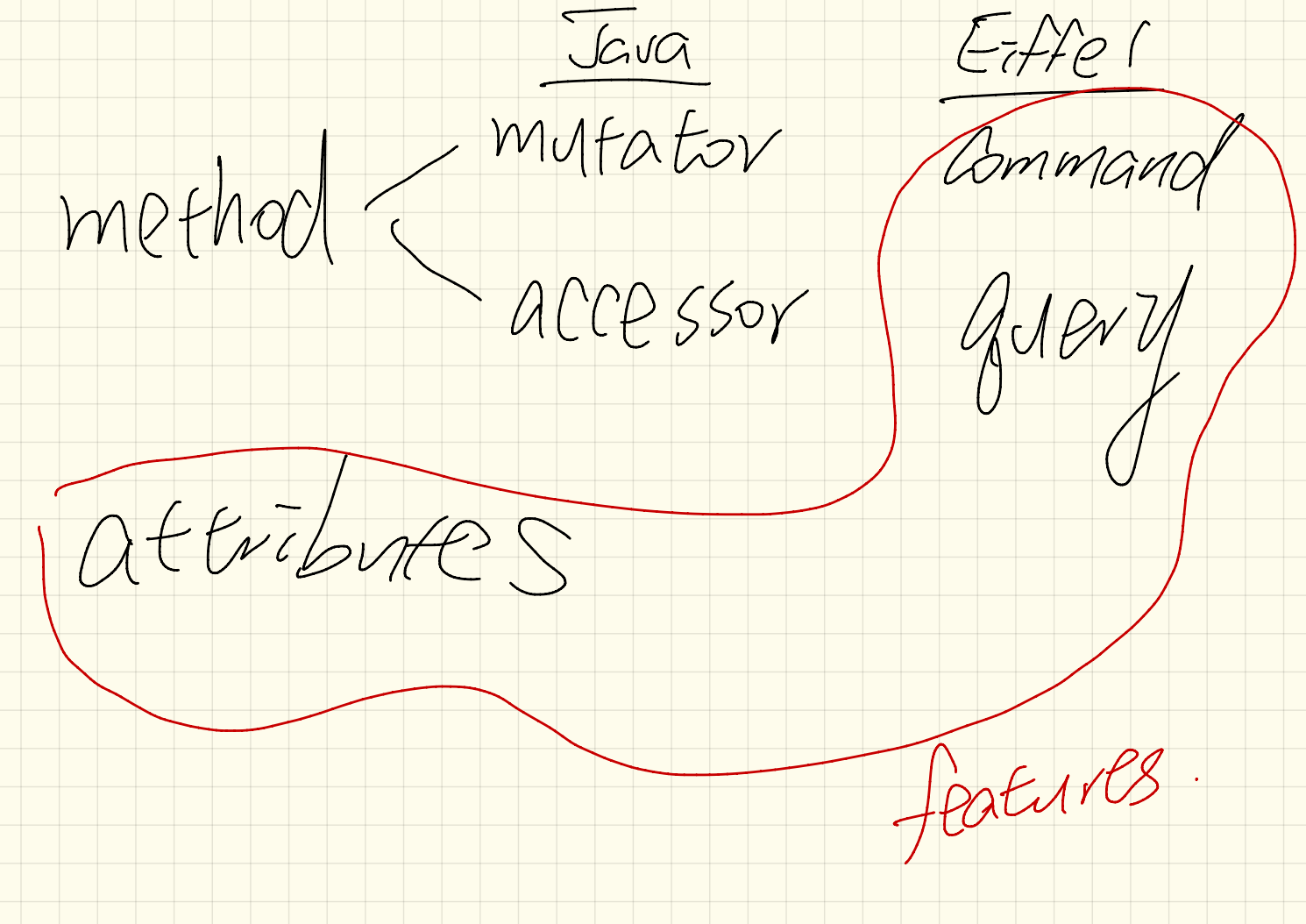
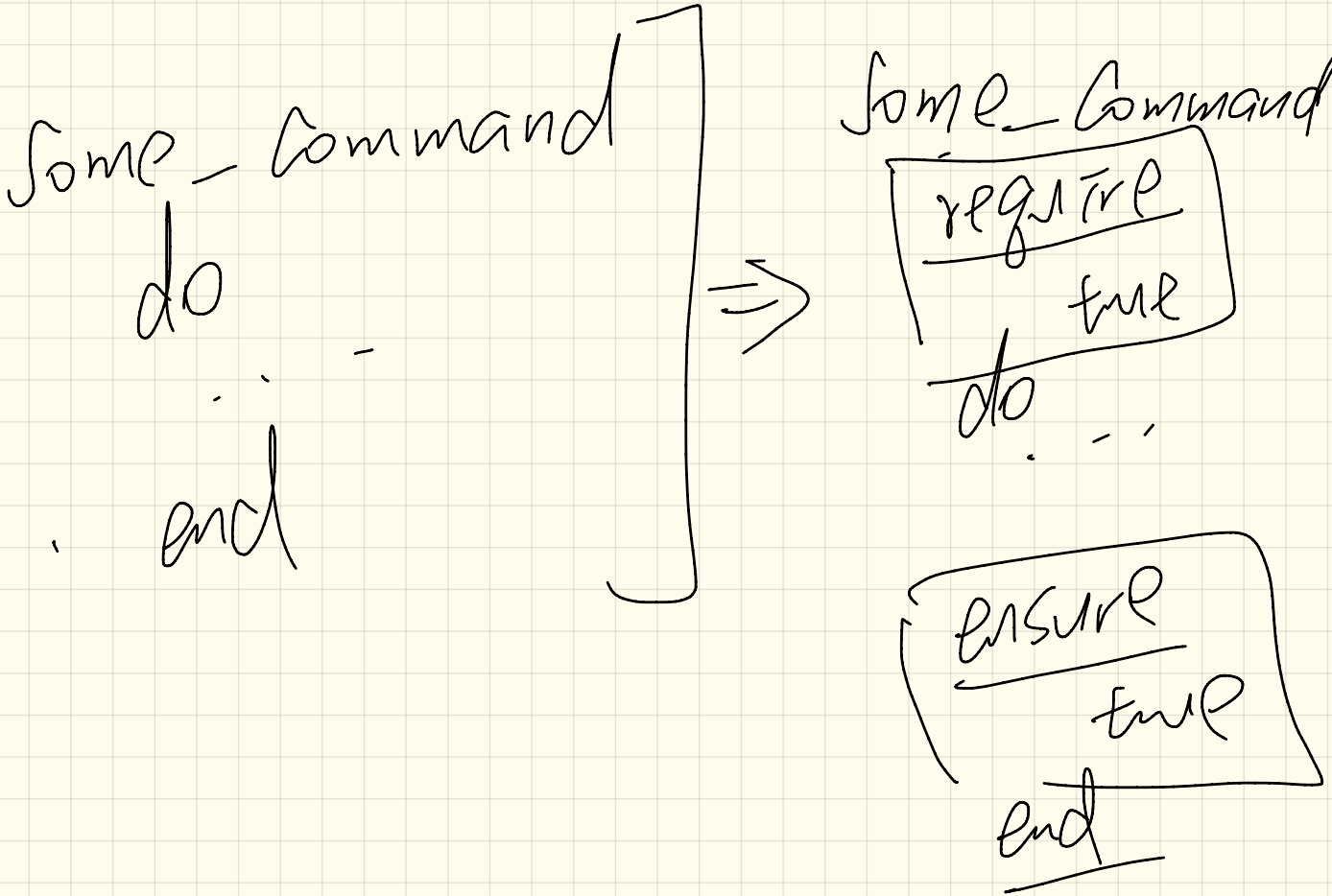
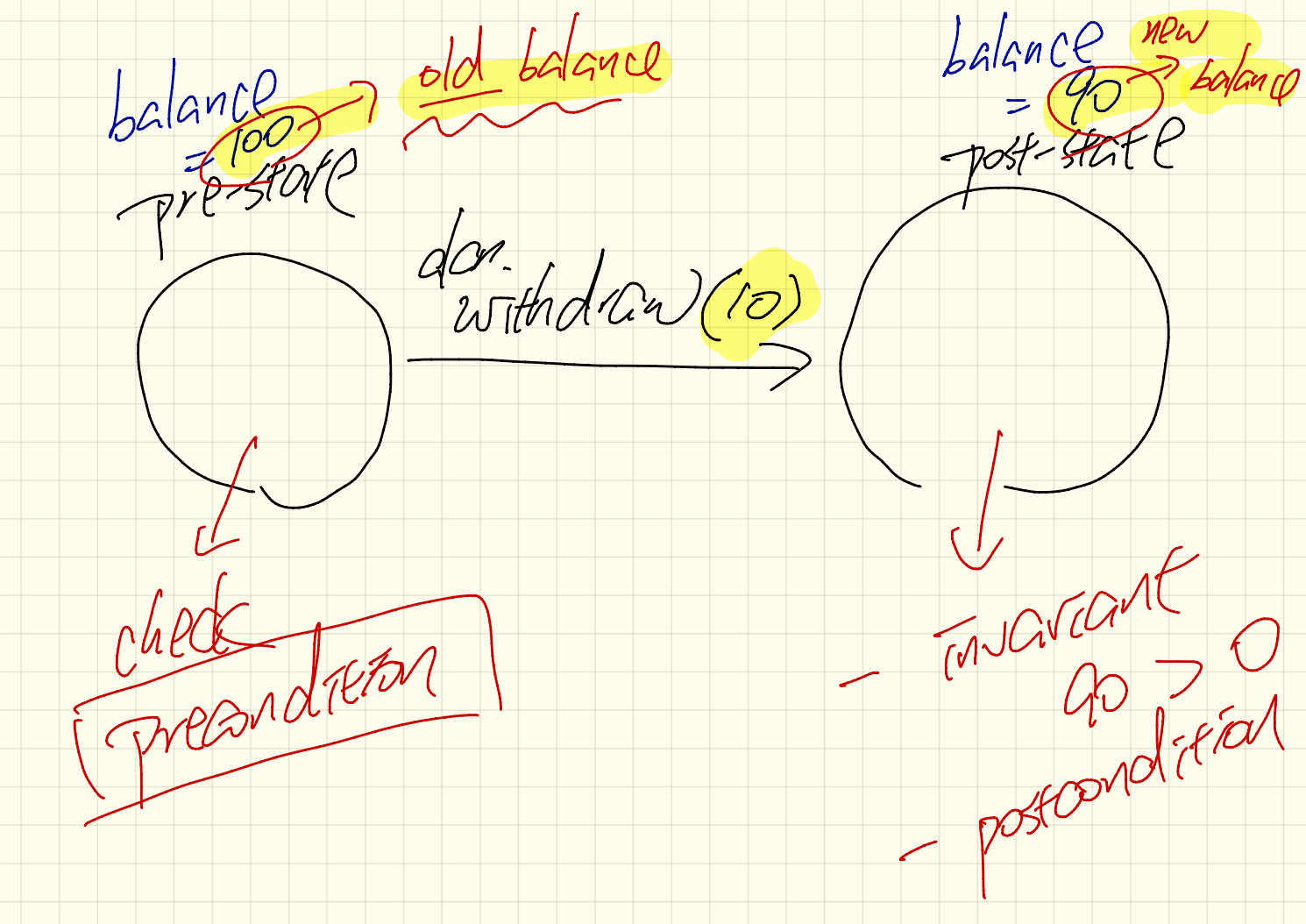
Lecture 3

Monday Sept. 18

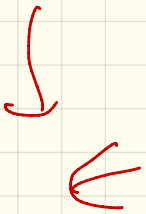

Java
Eiffel
method
mutator
accessor
command
query
attributes
features.


some_command
do
...
end
⇒
some_command
require
true
do
...
ensure
true
end


balance = 100
pre-state
old balance
do_withdraw(10)
balance = 90
post-state
new balance
check precondition
- invariant
90 > 0
- postcondition

int i

i : INTEGER



object creation

Account alan = new Account ("alan", 10);

Eiffel

create {ACCOUNT} alan.
make ("alan", 10)

void m() { ... }

---

m()

db

...

end

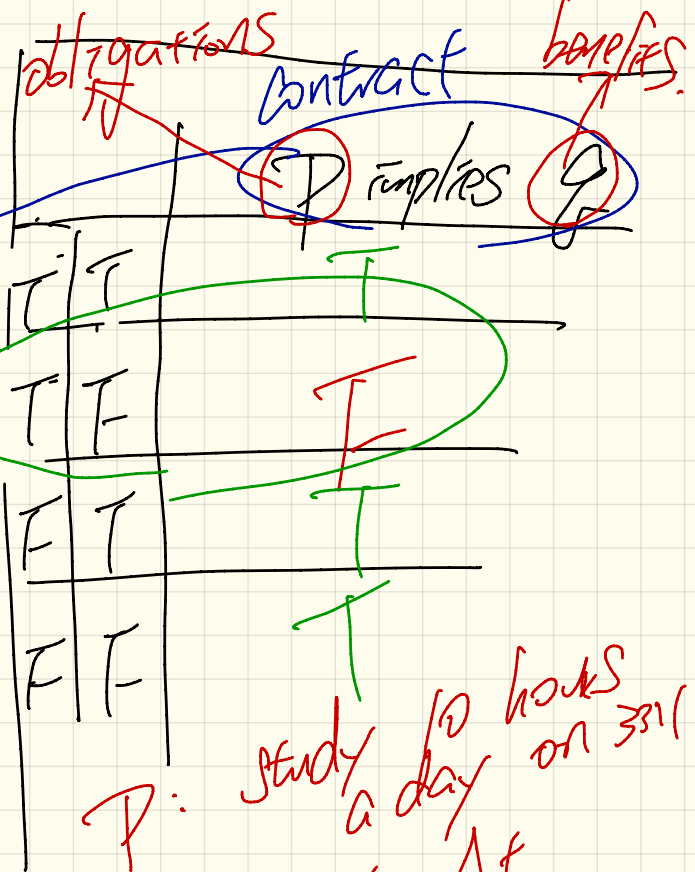
Logical Operators

$\wedge$ and

$\vee$ or

$\Rightarrow$ implies

$\neg$ not



$P \Rightarrow Q$ is True if contract is not violated.

P: study 10 hours a day on 331

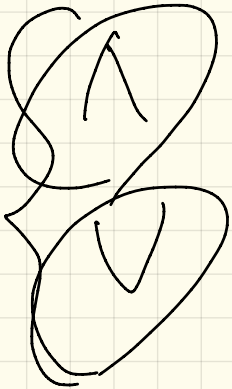
Q: get A+

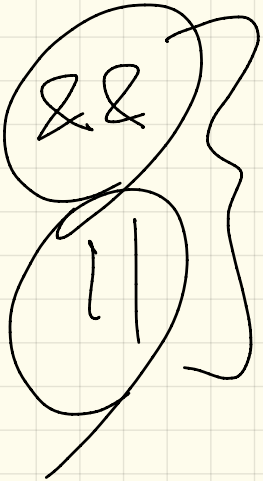
In Java SCE

E_1 && E_2

⇒ no F don't care

¬ !





not quite the same ! short-circuit effect -

SCE.

int a[]

&& ≠ ∧

0 ≤ i && i < a.length a[i]
&&

Eiffel

and ∧

or ∨

without SCE

and then ≡ &&

or else ≡ ||

Math

~~Array is sorted~~

a : ~~ARRAY~~ [INTEGER]

a.lower a.upper

$\forall i : \text{INTEGER} \mid a.lower \leq i \leq a.upper \bullet$

$a[i] \leq a[i+1]$

Eiffel

across

a.lower |..| (a.upper − 1)

a list integers.

as

i

all

some

i.item

cursor

i.item

end

a[i] <= a[i+1]

across

1 | 1..| 10 as i

all some

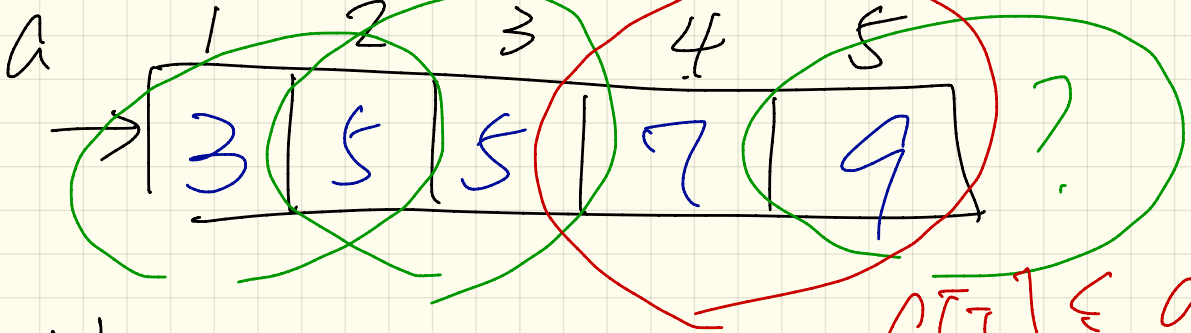
i. item \ 2 = 0

end

boolean expression

false

true



$a[i] \leq a[i+1]$

$\forall i : \text{INTEGER} \mid 1 \leq i \leq 5 \bullet$

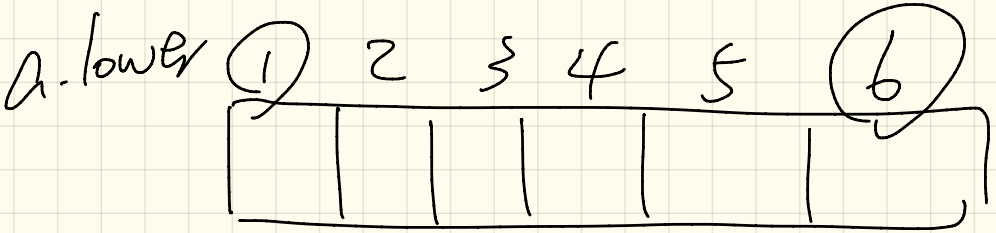
$5 - 1$

$a[i] \leq a[i+1]$

$a[1] \leq a[2]$ $\quad a[5] \leq a[6]$

$a[2] \leq a[3]$

$\vdots$



Size: a.upper − a.lower + 1

create a.make_empty

a.lower = 1

a.upper = 0

0 − 1 + 1 = 0