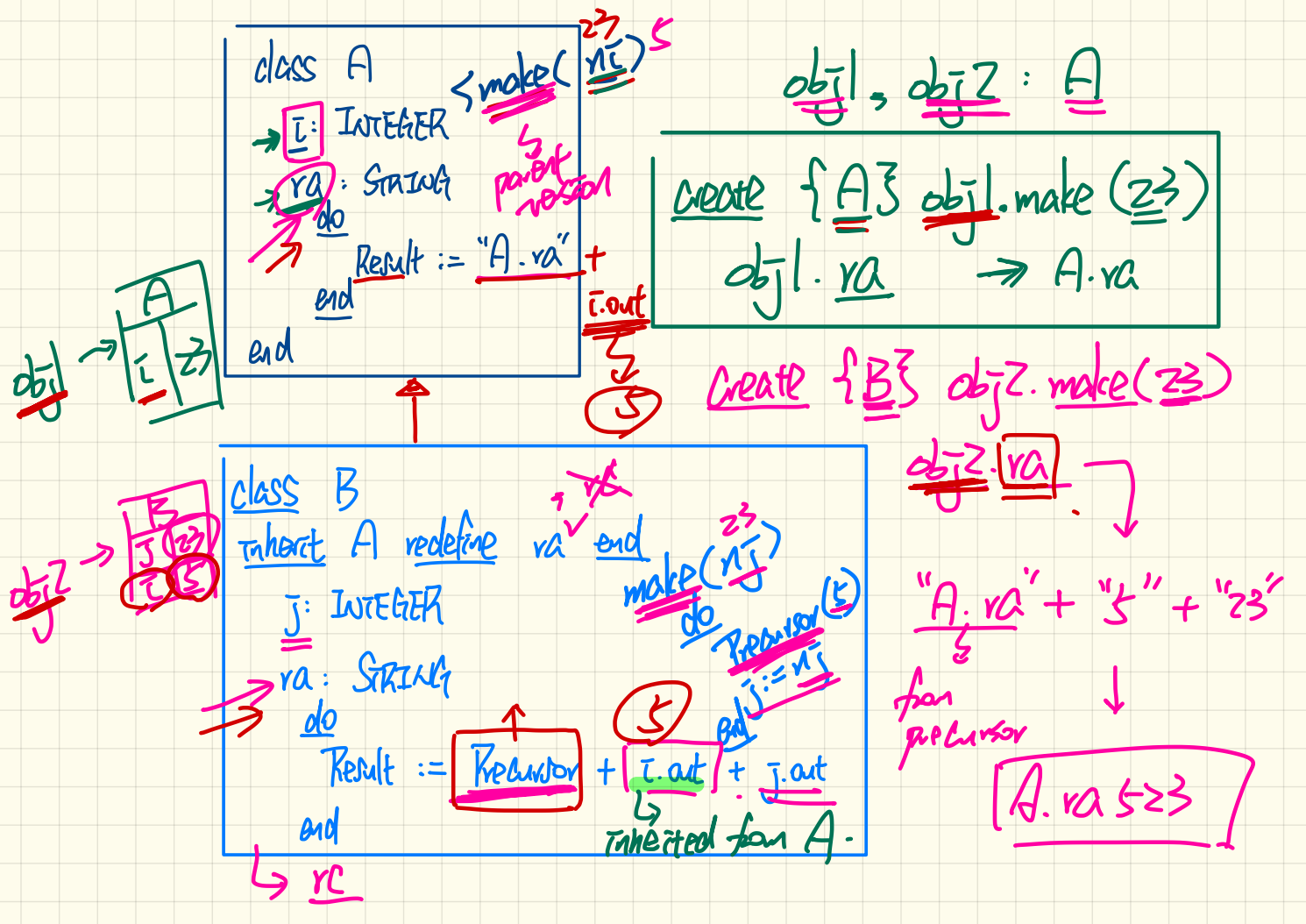
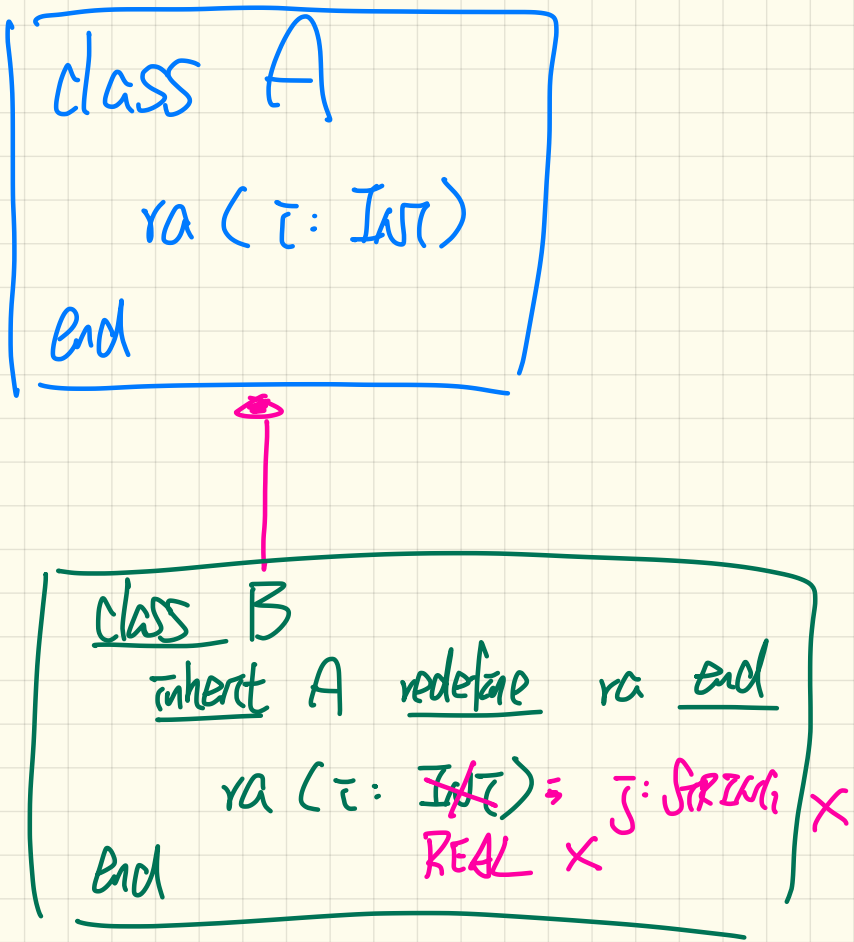
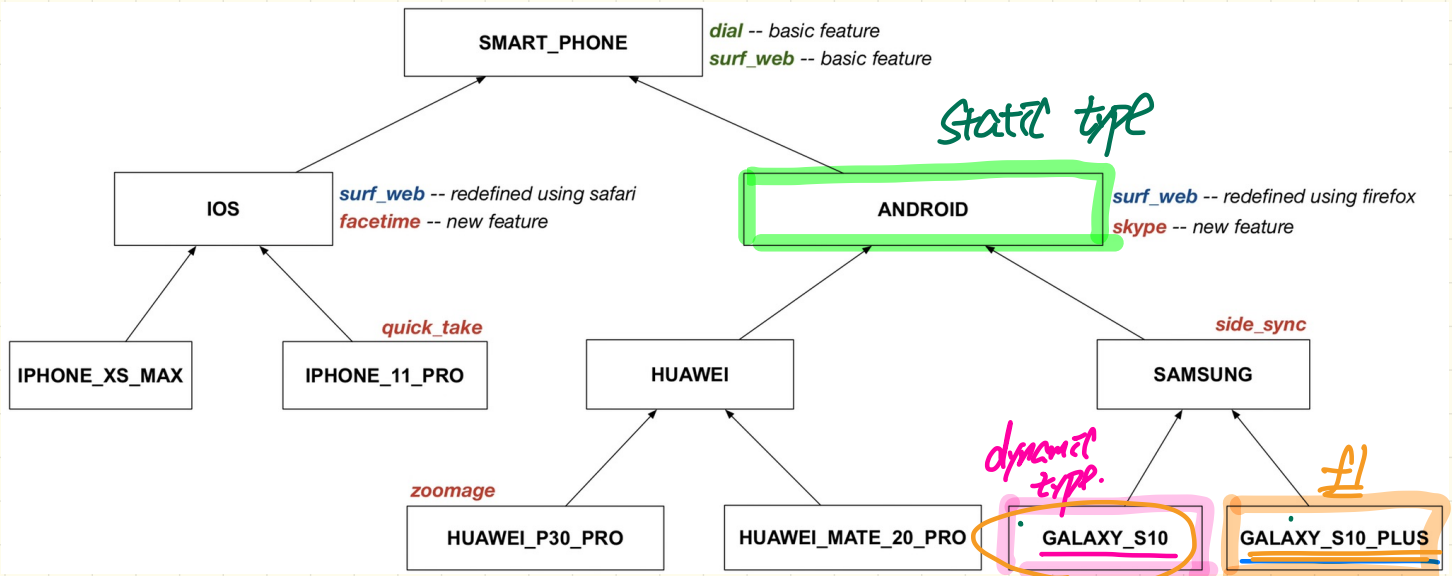
EECS3311 Software Design (Fall 2020)

Q&A - Lecture Series W7

Tuesday, November 3


class A
i: INTEGER
ra: STRING
do
Result := "A.ra" + i.out
end
end
make(ni)
parent version
obj1, obj2 : A
create {A} obj1.make(23)
obj1.ra → A.ra
create {B} obj2.make(23)
obj2.ra
class B
inherit A redefine ra end
j: INTEGER
ra: STRING
do
Result := Precursor + i.out + j.out
end
make(ni, nj)
do
Precursor
i := ni
j := nj
end
"A.ra" + "5" + "23"
from precursor
A.ra523
inherited from A.


class A
ra (i: Int)
end
class B
inherit A redefine ra end
ra (i: ~~Int~~ REAL x; j: STRING x)
end



mine : ANDROID

create {GALAXY_S10} mine.make(...)

cast : check attached {G_S10_PLUS} mine as n_p then

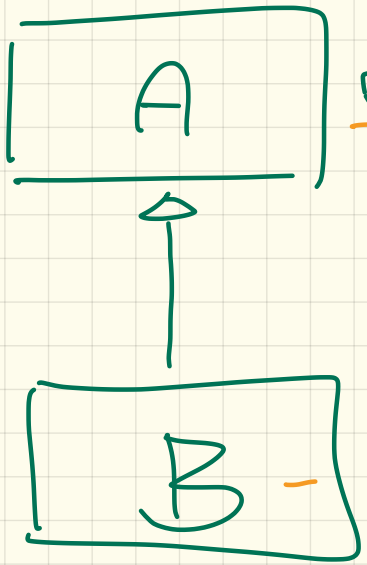
boolean expression

False.

No

have same expect

true : at the moment G_S10 and G_S10_P ?



f1

However, even though right now both A and B can both support f1, the compiler does not allow A for substituting B. (∵ B might be extended later)

Can B satisfy the expectation of A?

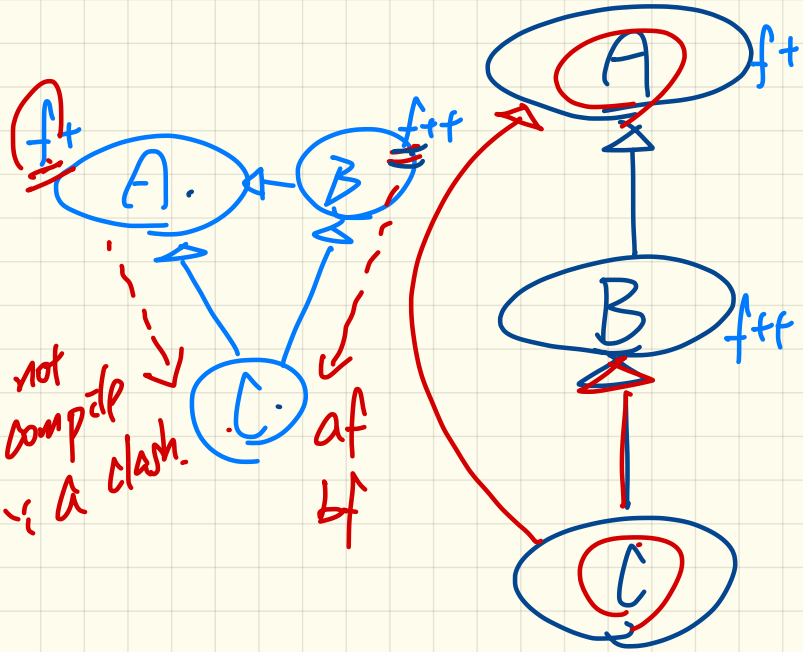
YES ∵ B is a descendant class of A

oa : A

ob : B

oa := ob ✓

ob := oa ✗



class C extends B ~~A~~

class C implements B, A ✓

↳ multiple inheritance
(partially supported in Java)

```
class C
    inherit A rename f as af end
            B rename f as bf end.
```