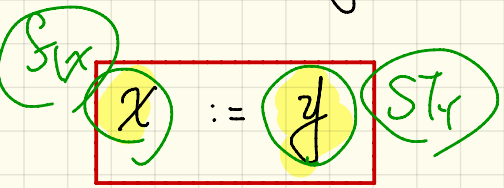
Thursday Oct. 18
Lecture 11

- Review Session

Friday 3pm LAS B

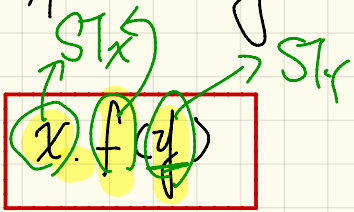
Type Checking Rules (1)



s1 : STUDENT
s2 : RS
s3 : NRS

1 s1 := s2 ✓
2 s1 := s3 ✓
3 s2 := s1 ✗
4 s3 := s1 ✗
5 ✓ s2 := s3 ✗
6 s3 := s1 ✗

Type Checking Rules (2)




```
class SMS
  add_rs(s : RS)
    do
    end ...
end
```

sms : SMS

s1 : STUDENT

s2 : RS

s3 : NRS

STUDENT

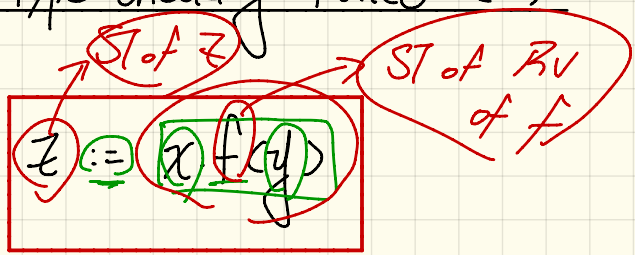
1 ✓ s1.add_rs(s2)

2 ✗ sms.add_rs(s1) → ST : S

3 ✓ sms.add_rs(s2) → ST : RS

4 ✗ sms.add_rs(s3) → NRS

Type Checking Rules (3)




```
class SMS
    get_s(i: INTEGER): STUDENT
    do
        ...
    end
end
```

sms: SMS

s1: STUDENT

s2: RS

s3: NRS

1 s1 := s2.get_s(1) ✗

2 s1 := sms.get_s(1) ✓

3 s2 := sms.get_s(1) ✗

4 s3 := sms.get_s(1) ✗

NRS STUDENT

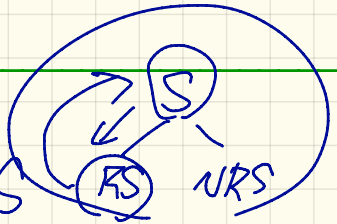
Type Checking Rules (4)

```
check attached {C} y then
    ...
end
```

STy

C is [ancestor of STy
       descendant of STy]

```
class SMS
    get_S (i: INTEGER): STUDENT
    do
        ...
    end
end
```



sms : SMS
s1 : STUDENT
s2 : RS
s3 : URS

1 check attached {RS} s1 then ... end ✓
2 check attached {STUDENT} s2 then ... end ✓
3 check attached {SMS} s1 then ... end ✗
4 check attached {RS} s3 then ... end ✗
5 check attached {RS} sms.get(1) then ... end ✓

STUDENT

Type Checking Rules (5)

```
check attached {C} y as temp then
    x := temp
end
```

ST: C

ST_x

ST: C

```
class SMS
    get_s(i: INTEGER): STUDENT
    do
        ...
    end
end
```

```
check attached {RS} sms.get_s(1) as temp then
    s1 := temp
    s3 := temp
end
```

S

ST: RS

sms: SMS

s1: STUDENT

s2: RS

s3: NRS

s1 := temp ✓ RS

2 s3 := temp ✗

NRS ✗ RS

Type Checking Rules (6)

```
check attached {C} y as temp then
    x.f(temp)
end
```

ST: C

```
class SMS
    get_s(i: INTEGER): STUDENT
    do
        ...
    end
    add_rs(s: RS)
    do
        ...
    end
```

```
RS temp = (RS) sms.get_s(1);
sms.add_rs(temp);
```

sms: SMS

s1: STUDENT

s2: RS

s3: NRS

① compiles

```
check attached {RS} sms.get_s(1) as temp then
    sms.add_rs(temp)
end
```

RT. S ST: RS

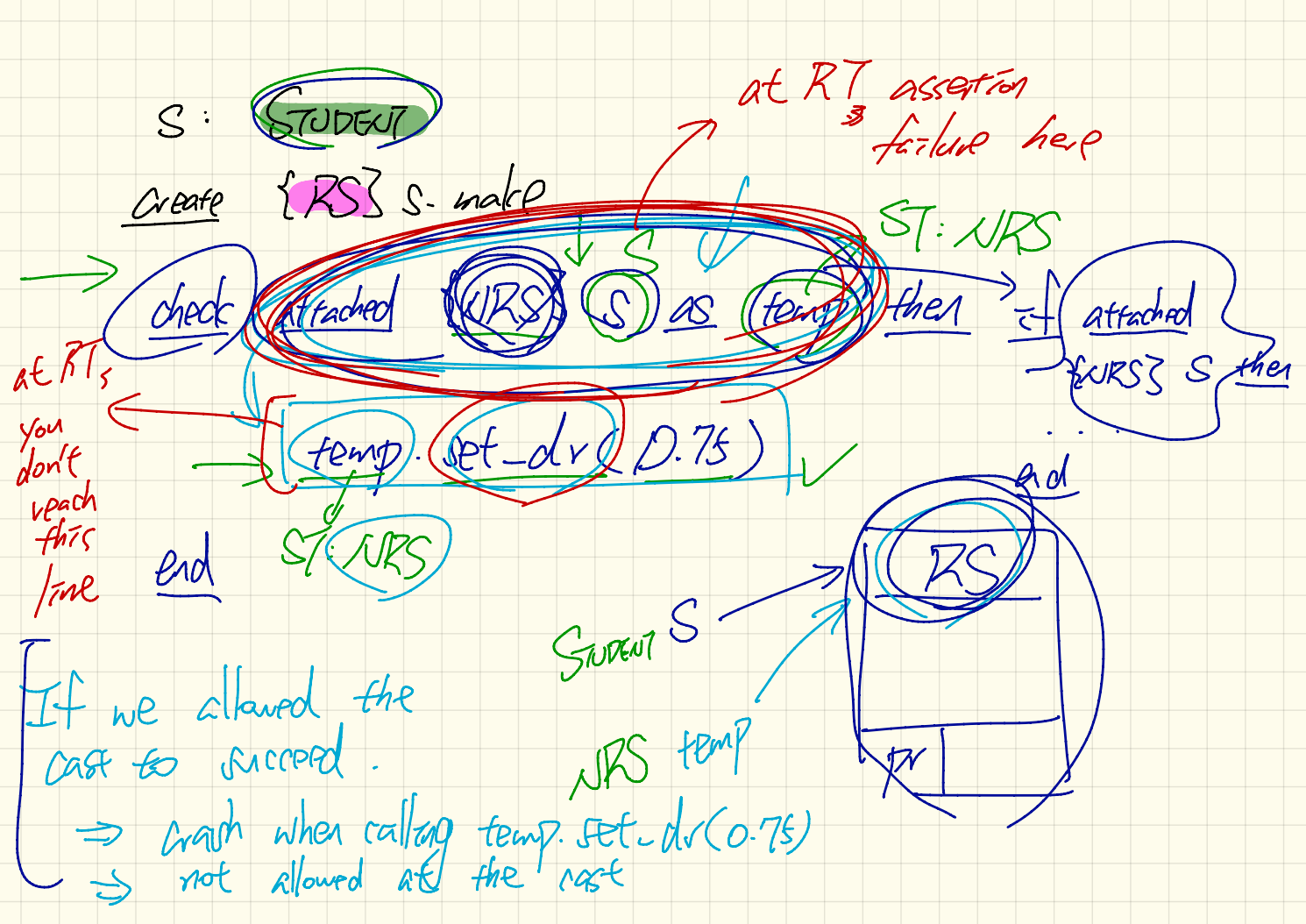
DT: NRS

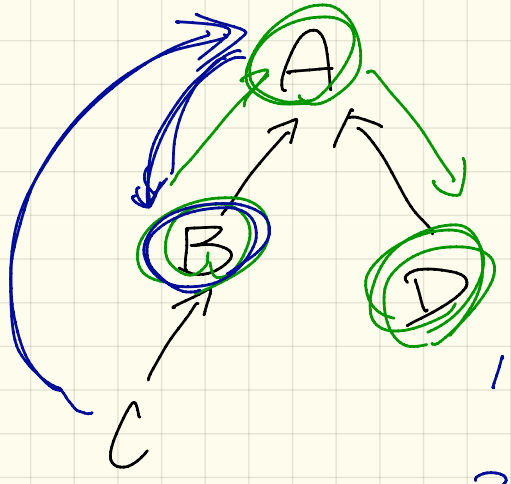
temp.set_pr

② ST: NRS

```
check attached {NRS} sms.get_s(1) as temp then
    sms.add_rs(temp)
end
```

NRS ✗


S : STUDENT
create {RS} s.make
at RT assertion failure here
ST: NRS
check attached {NRS} S as temp then
attached {NRS} S then
temp.set_dr(0.75)
at RT, you don't reach this line
end
ST: NRS
end
RS
dr
STUDENT S
NRS temp
If we allowed the cast to succeed.
⇒ crash when calling temp.set_dr(0.75)
⇒ not allowed at the cast



b: B

d: D

c: C

1 create {C} b.make

2 check attached {D} b as ST: B temp then

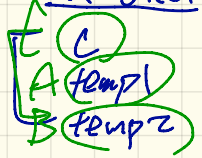
3 d := temp

end

check attached {A} c as temp1 A then B

check attached {B} temp1 as temp2 then

end

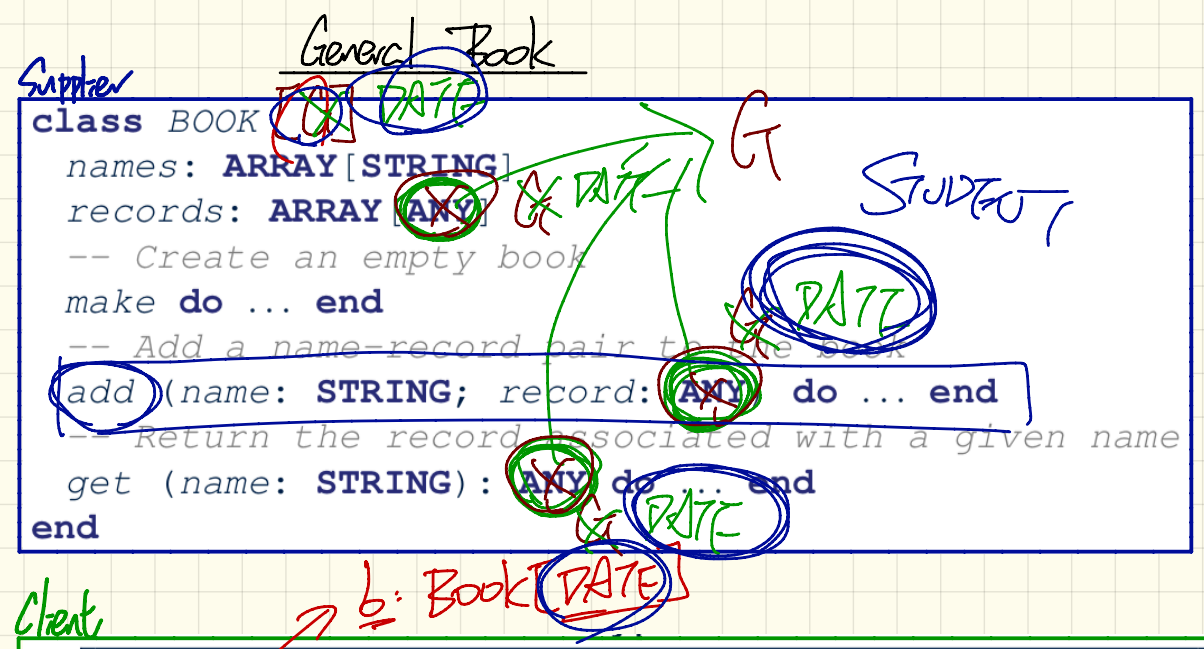


check attached {A} b as temp 1 and then

attached {B} temp1 as temp2 then

:

end




```
1  birthday: DATE; phone_number: STRING
2  b: BOOK; is_wednesday: BOOLEAN
3  create {BOOK} b.make
4  phone_number := "416-677-1010"
5  b.add ("SuYeon", phone_number)
6  create {DATE} birthday.make(1975, 4, 10)
7  b.add ("Yuna", birthday)
8  is_wednesday := b.get("Yuna").get_day_of_week = 4
```