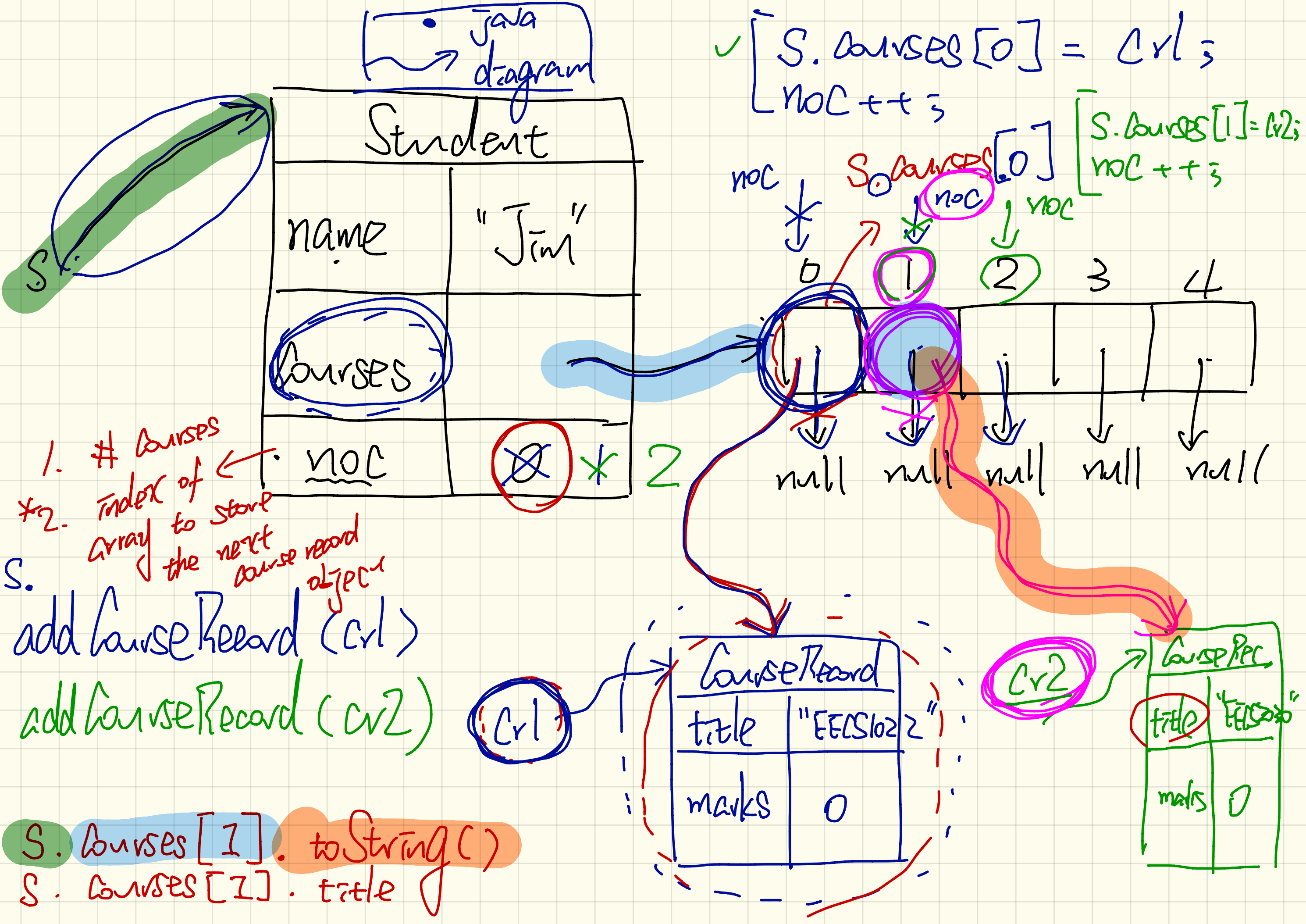
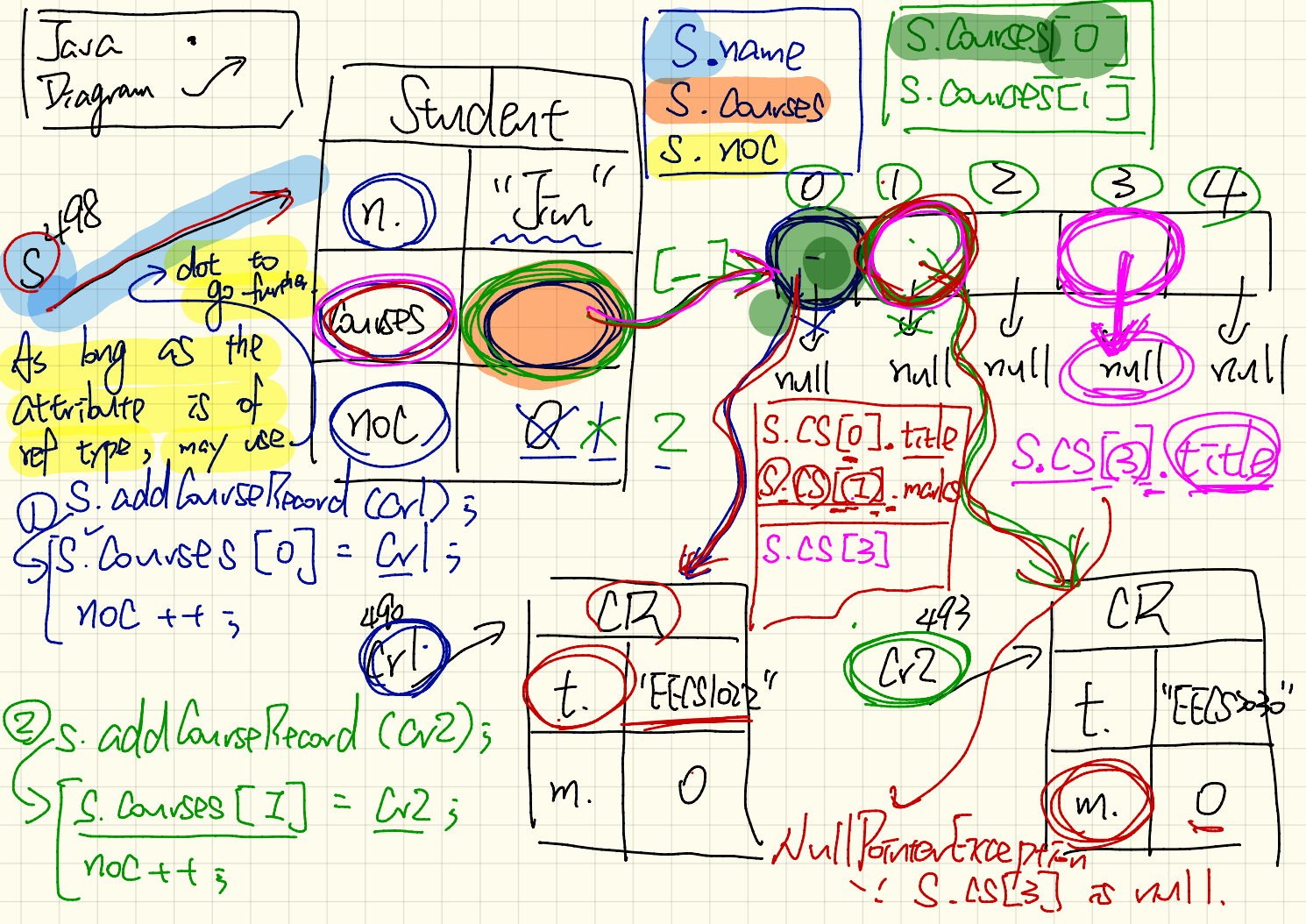
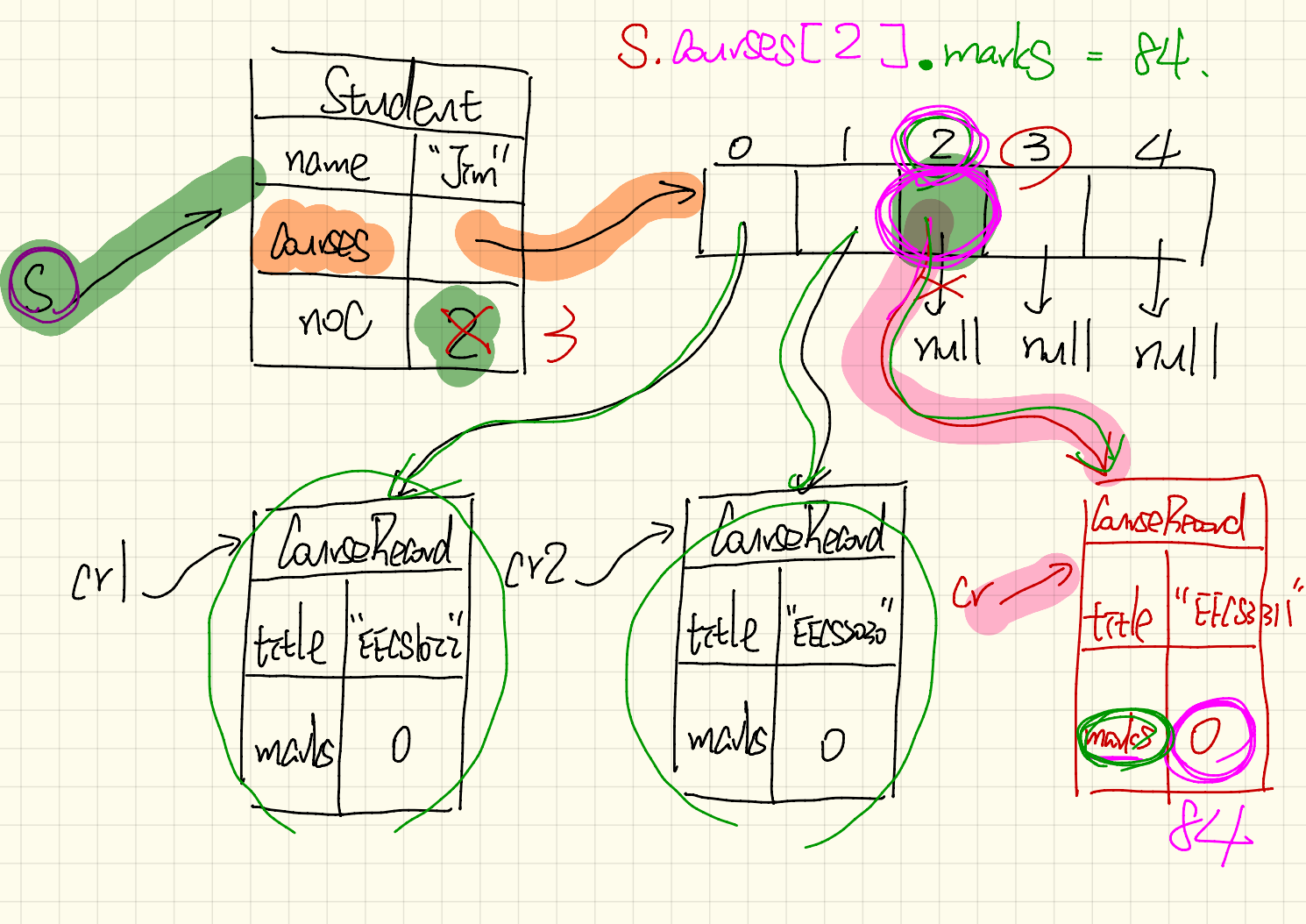
Model Development Pattern:

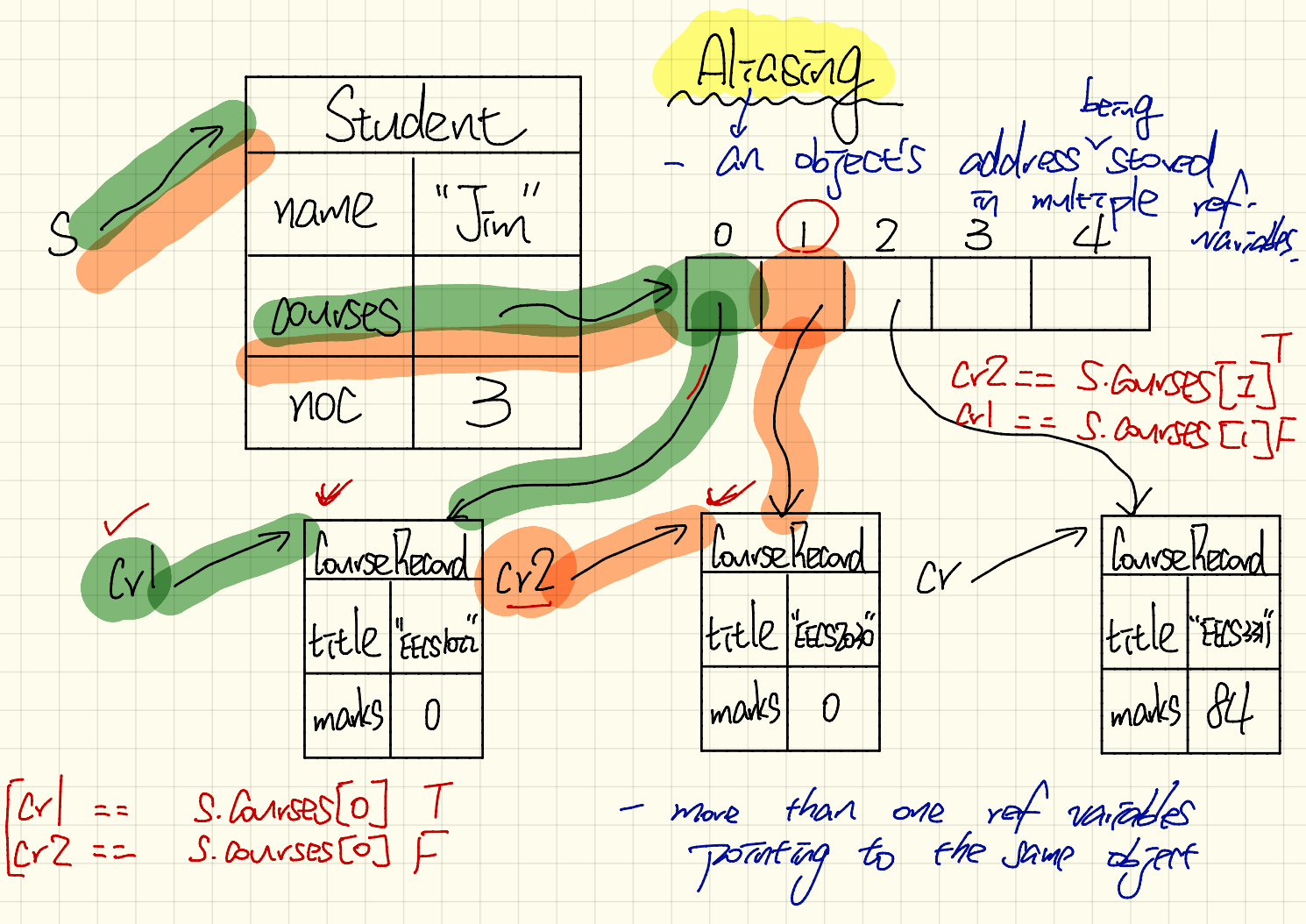
Attributes with Array-Referenced Types

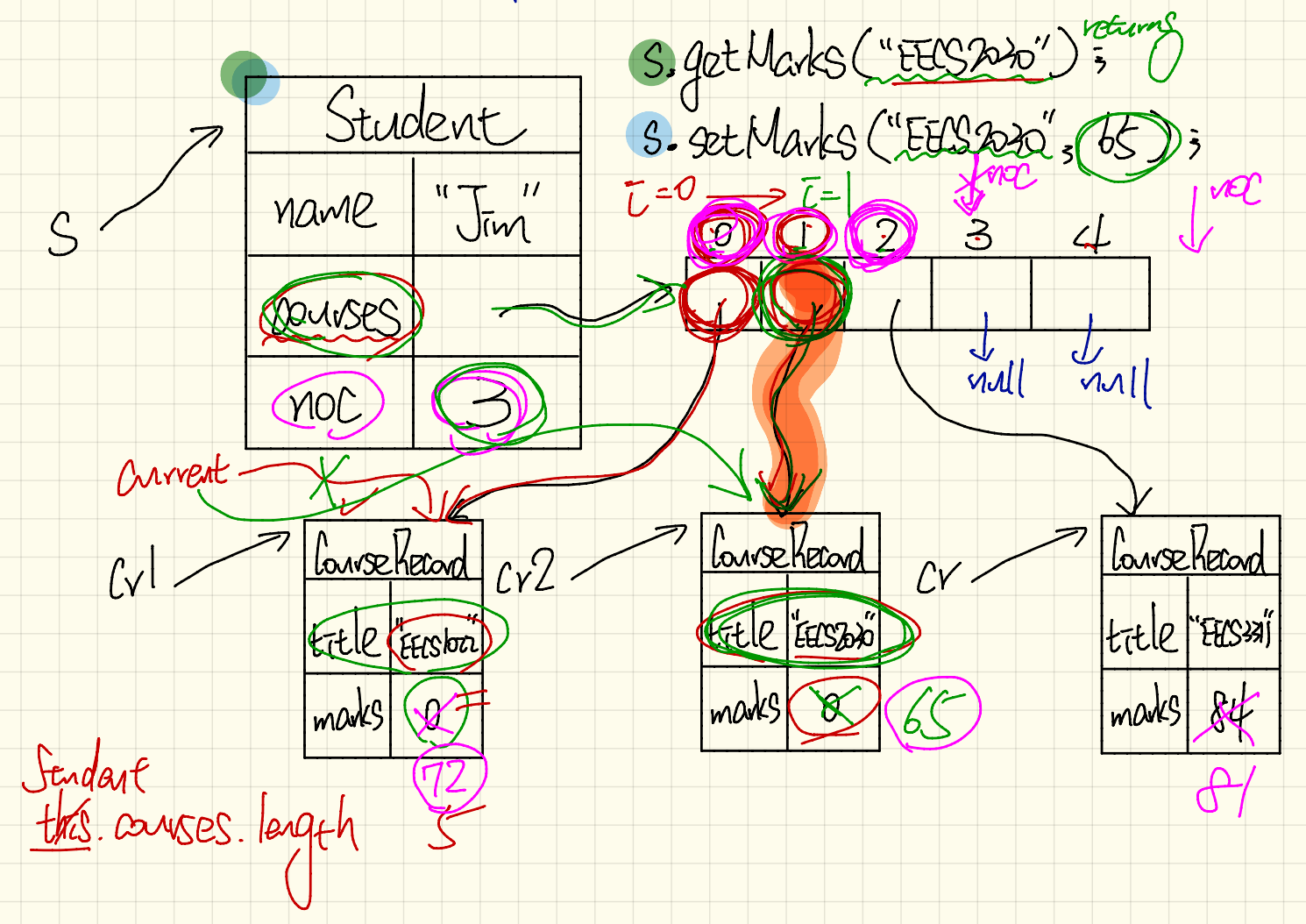


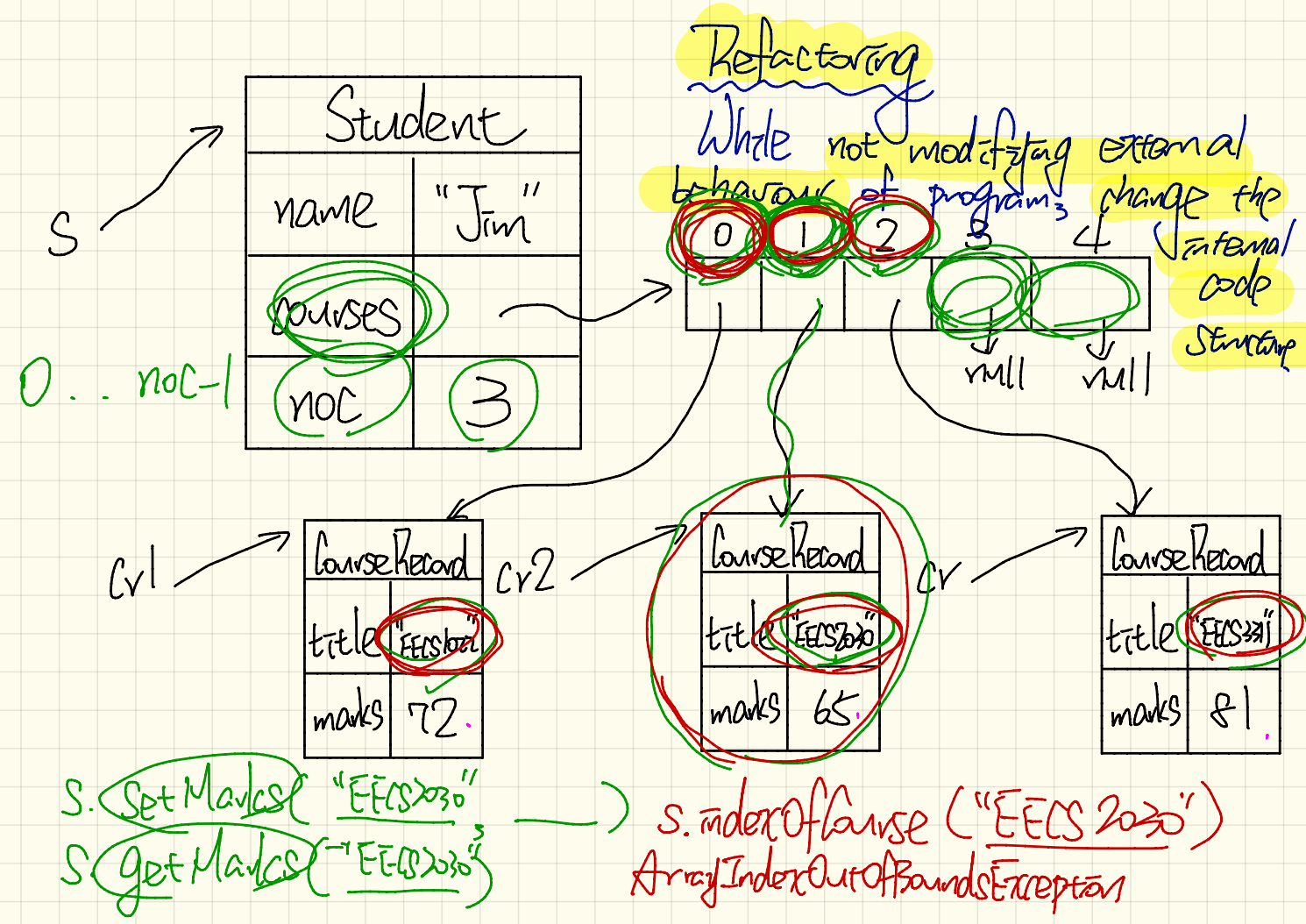
java diagram
Student
name
"Jim"
courses
noc
2
0 1 2 3 4
null null null null null
S.courses[0] = cr1;
noc++;
S.courses[1] = cr2;
noc++;
S.courses[0]
noc
1. # courses
*2. index of array to store the next course record object
S.
addCourseRecord (cr1)
addCourseRecord (cr2)
cr1
cr2
CourseRecord
title
"EECS1022"
marks
0
CourseRec
title
"EECS3[illegible]"
marks
0
S.courses[1].toString()
S.courses[1].title

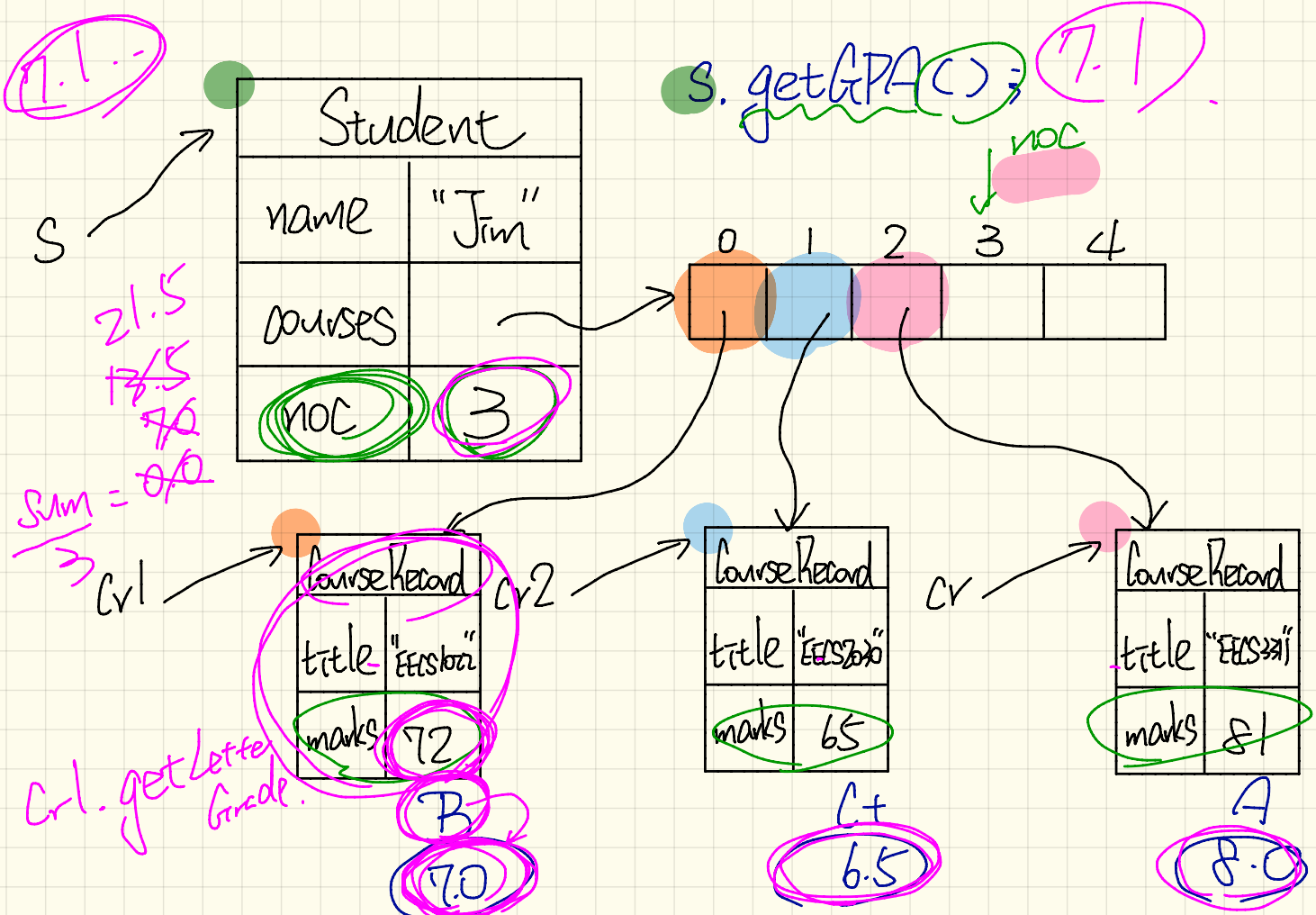

Java Diagram
Student
n.
"Jim"
courses
noC
0
S
498
dot to go further
As long as the attribute is of ref type, may use
S.name
S.courses
S.noC
S.courses[0]
S.courses[1]
0
1
2
3
4
null
null
null
null
null
S.CS[0].title
S.CS[1].marks
S.CS[3]
S.CS[3].title
2
① S.addCourseRecord(cr1);
S.courses[0] = cr1;
noC++;
496
cr1
CR
t.
"EECS102"
m.
0
② S.addCourseRecord(cr2);
S.courses[I] = cr2;
noC++;
493
cr2
CR
t.
"EECS203"
m.
0
NullPointerException
∵ S.CS[3] is null.

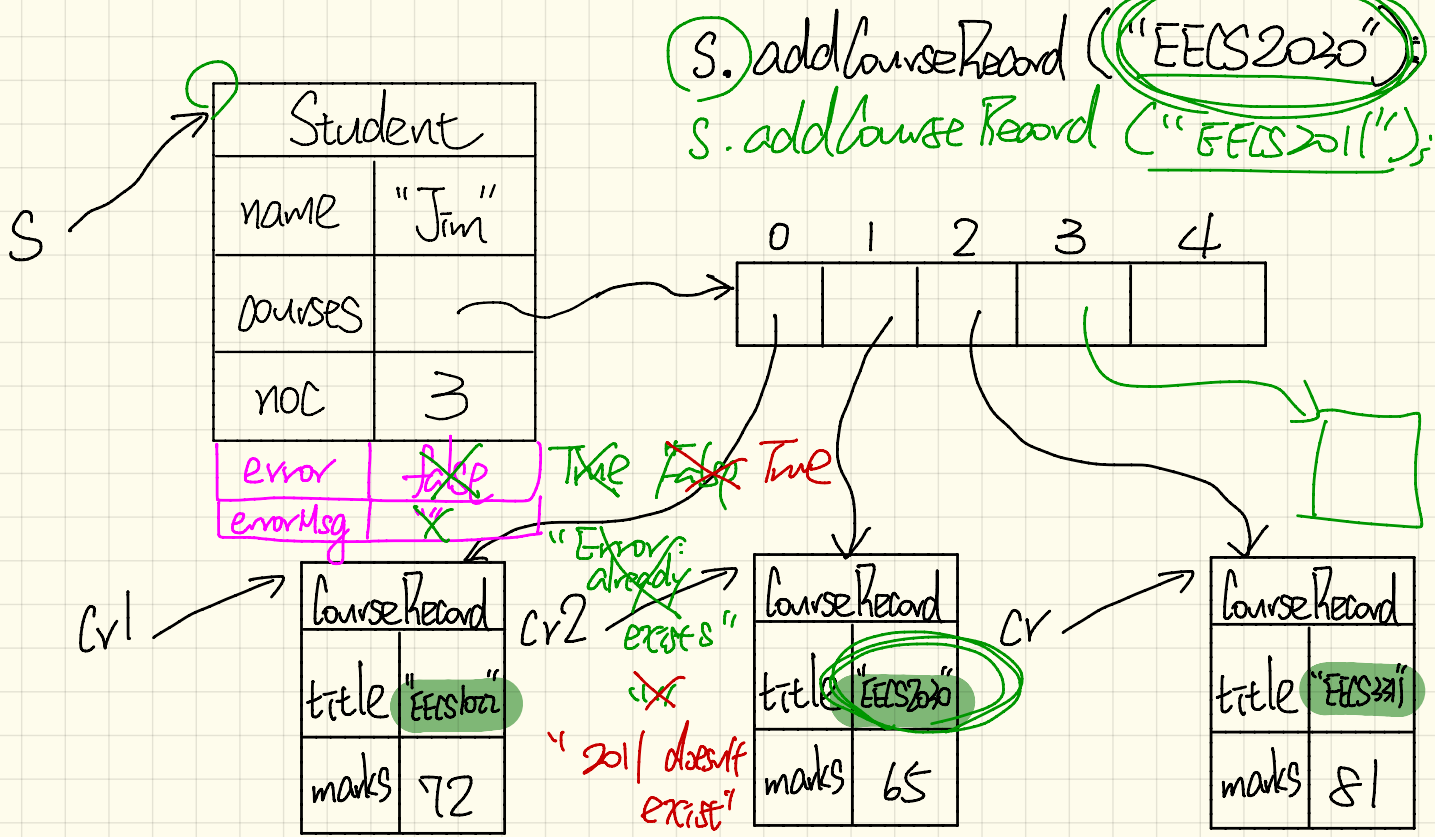

S.courses[2].marks = 84.
Student
name
"Jim"
courses
noC
2
3
S
0
1
2
3
4
null
null
null
cr1
courseRecord
title
"EECS1022"
marks
0
cr2
courseRecord
title
"EECS2030"
marks
0
cr
courseRecord
title
"EECS3311"
marks
0
84

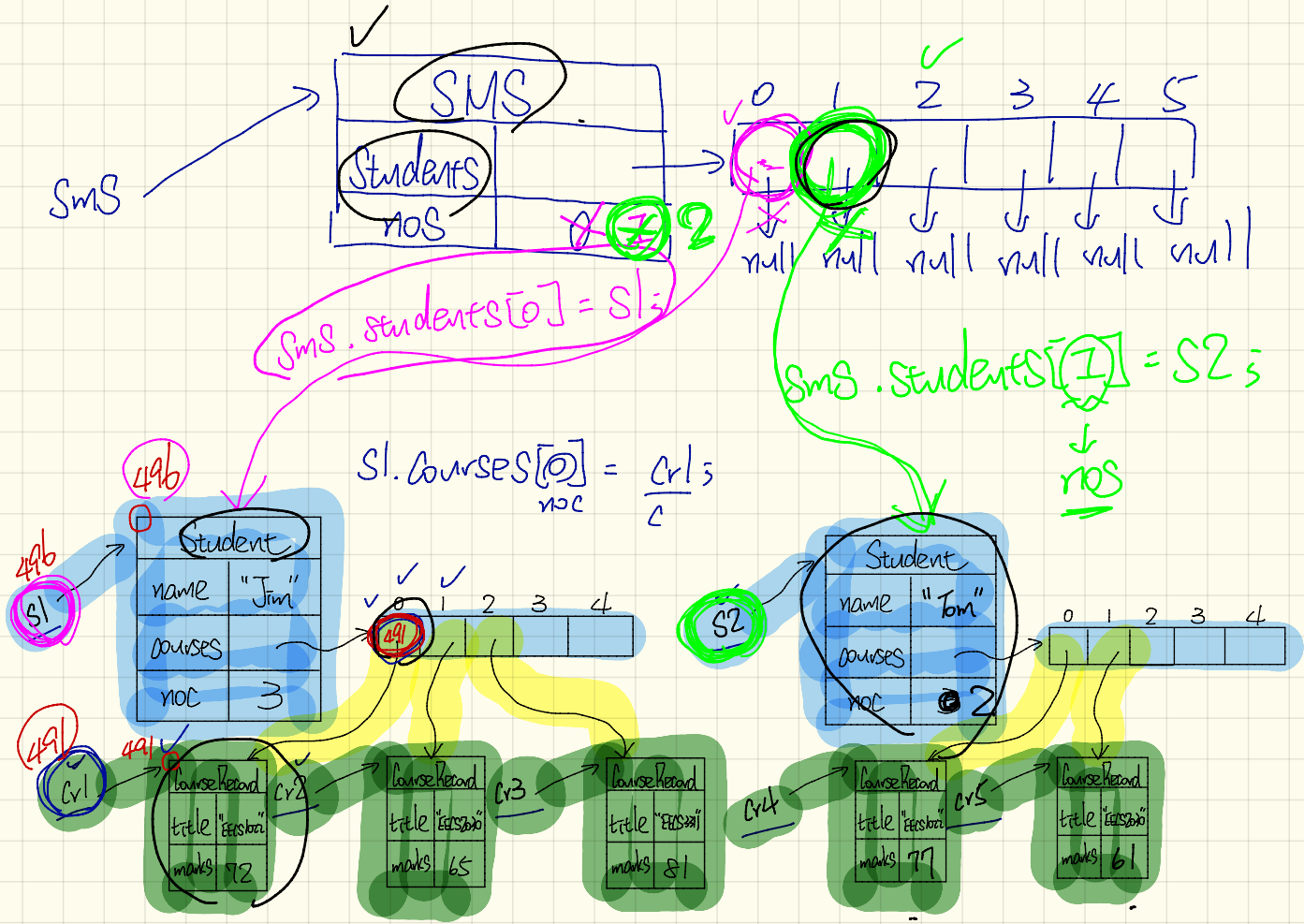

Aliasing
- an object's address stored in multiple ref. variables being
Student
name
"Jim"
courses
noc
3
s
0
1
2
3
4
cr2 == s.courses[1] T
cr1 == s.courses[1] F
cr1
cr2
cr
CourseRecord
title
marks
0
84
[cr1 == s.courses[0] T
cr2 == s.courses[0] F
- more than one ref variables pointing to the same object

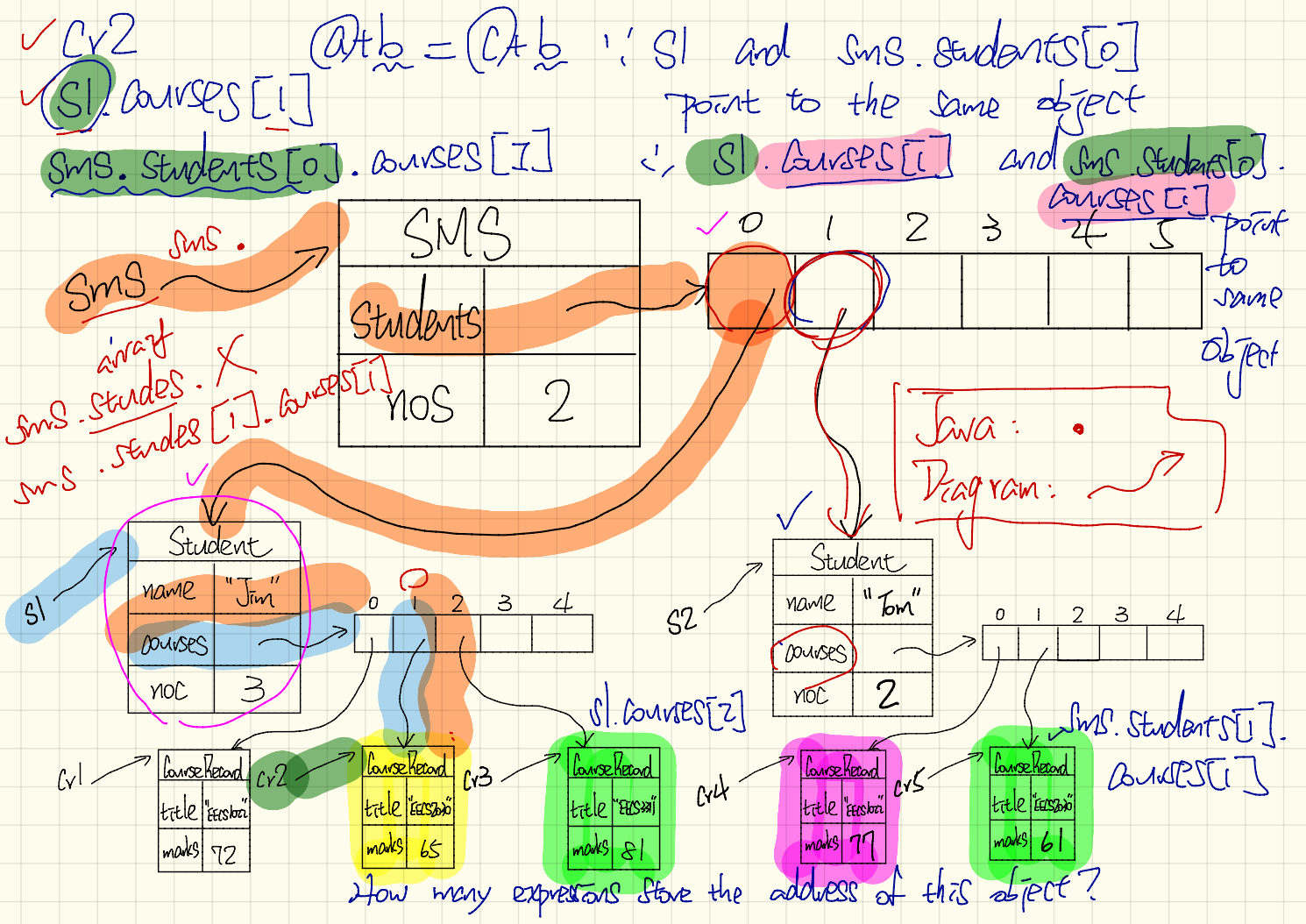

s.getMarks("EECS2030"); returns 0
s.setMarks("EECS2030", 65);
Student
name
"Jim"
courses
noc
3
s
current
i=0
i=1
0
1
2
3
4
noc
noc
null
null
cr1
cr2
cr
CourseRecord
title
"EECS1022"
marks
0
72
5
CourseRecord
title
"EECS2030"
marks
0
65
CourseRecord
title
"EECS3311"
marks
84
84
Student
this.courses.length

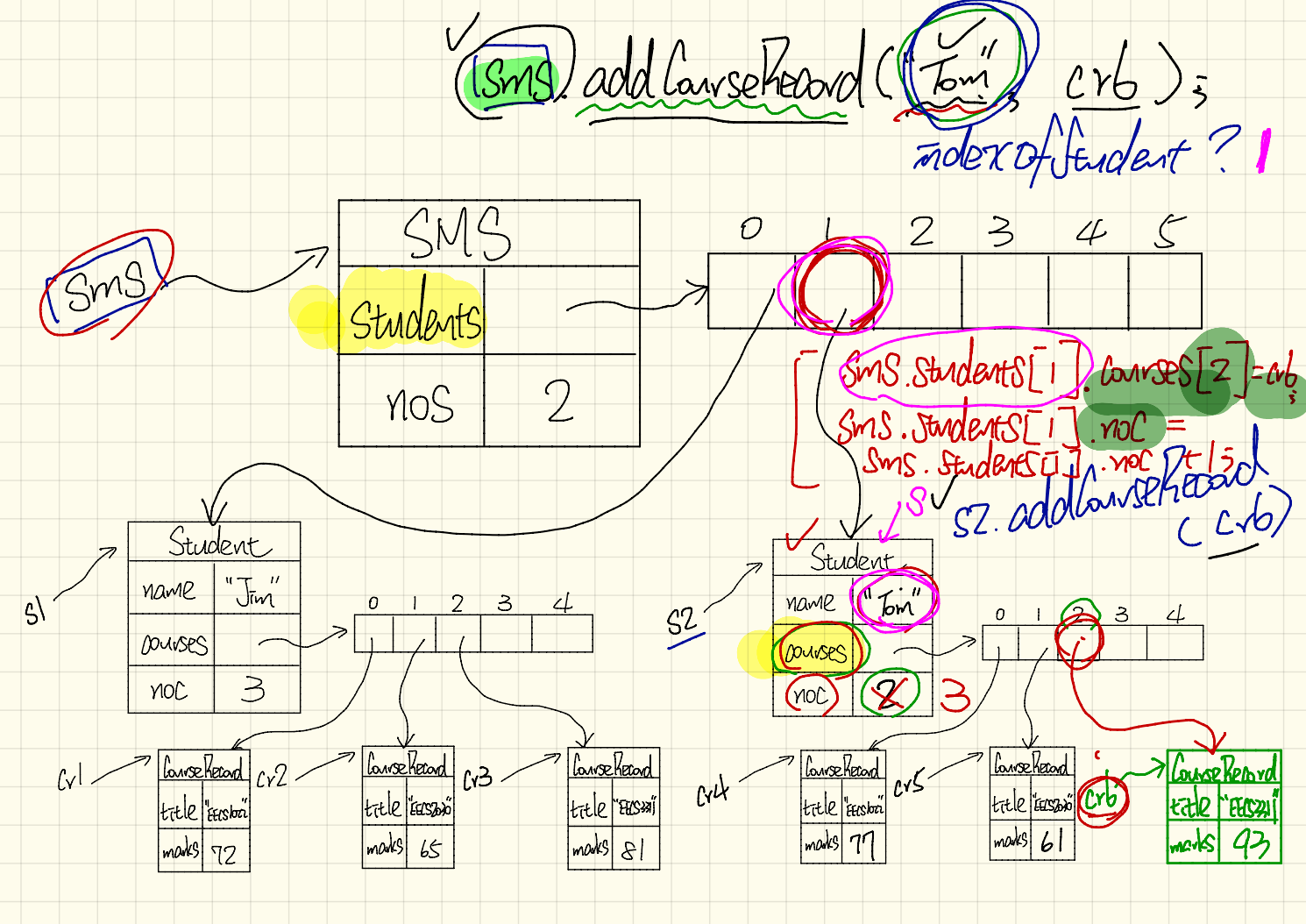

Refactoring
While not modifying external behaviour of program, change the internal code structure
Student
name
"Jim"
courses
noc
3
s
0 .. noc-1
0
1
2
3
4
null
null
CourseRecord
title
"EECS1021"
marks
72
cr1
CourseRecord
title
"EECS2030"
marks
65
cr2
CourseRecord
title
"EECS3311"
marks
81
cr
s.setMarks("EECS2030", 65)
s.getMarks("EECS2030")
s.indexOfCourse("EECS2030")
ArrayIndexOutOfBoundsException

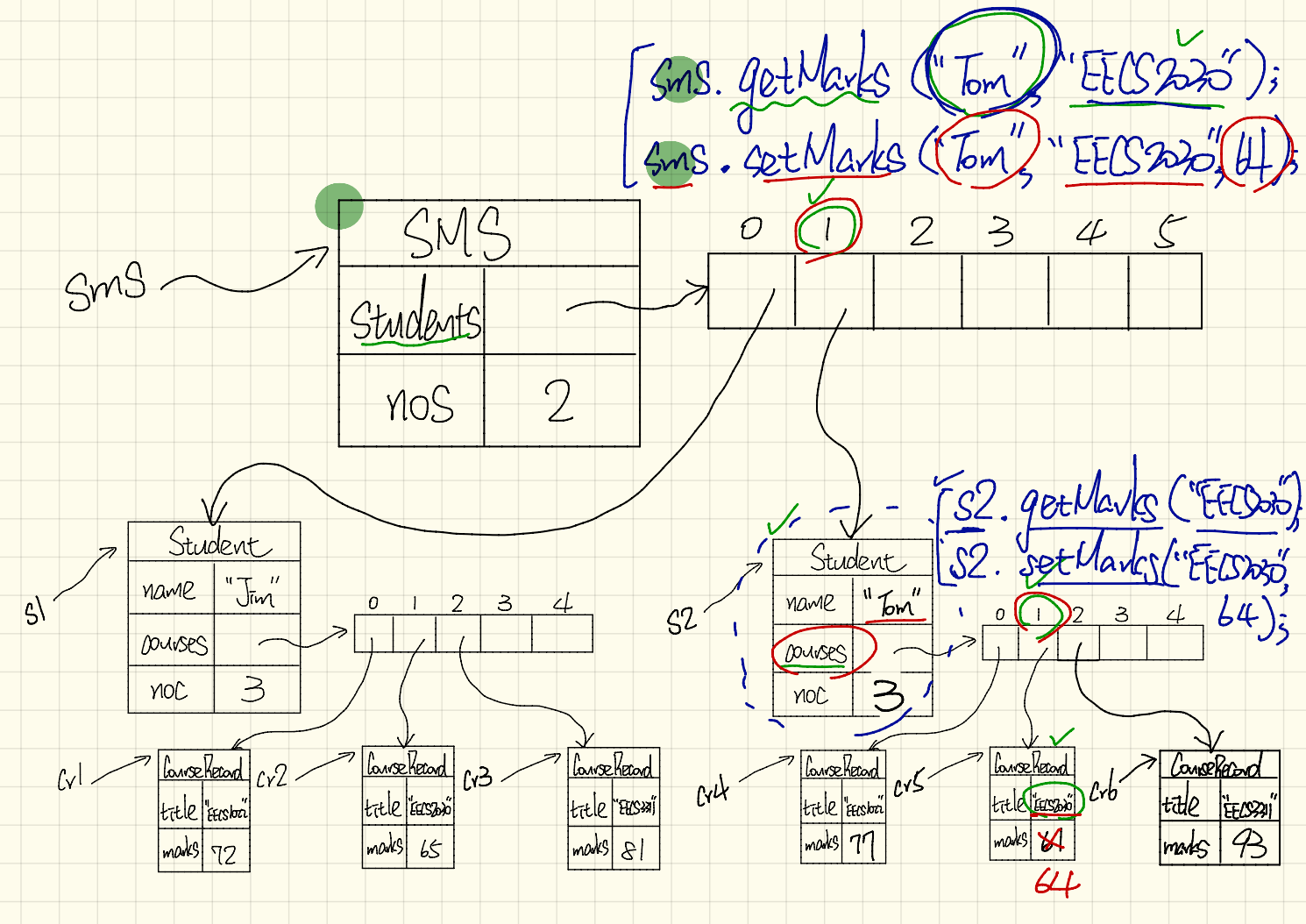

7.1
s.getGPA();
7.1
Student
name
"Jim"
courses
noc
3
s
noc
0
1
2
3
4
21.5
sum = 0.0
cr1
cr2
cr
CourseRecord
title
marks
72
65
81
B
C+
A
7.0
6.5
8.0
cr1.getLetterGrade();

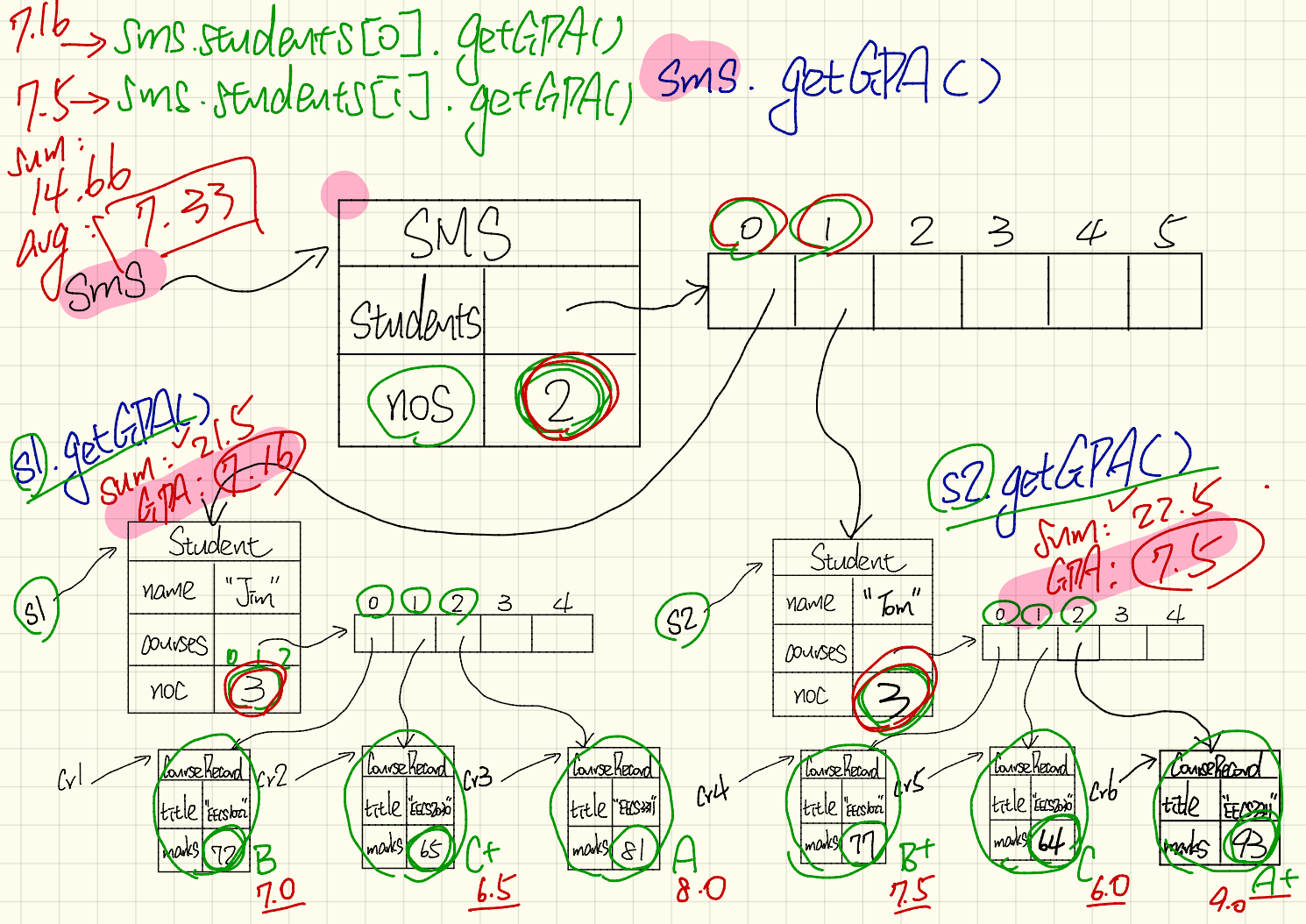

s.addCourseRecord("EECS2030");
s.addCourseRecord("EECS2011");
s
Student
name
"Jim"
courses
noc
3
error
false
errorMsg
""
0
1
2
3
4
True
False
True
"Error: already exists"
"2011 doesn't exist"
cr1
CourseRecord
title
"EECS1022"
marks
72
cr2
CourseRecord
title
"EECS2030"
marks
65
cr
CourseRecord
title
"EECS3311"
marks
81


SMS
Students
noS
SmS
0 1 2 3 4 5
null null null null null null
SmS.students[0] = S1;
SmS.students[I] = S2;
noS
S1.courses[0] = Cr1;
noC
496
Student
name "Jim"
courses
noC 3
S1
496
491
0 1 2 3 4
Cr1
491
CourseRecord
title "EECS281"
marks 92
Cr2
CourseRecord
title "EECS280"
marks 65
Cr3
CourseRecord
title "EECS370"
marks 81
S2
Student
name "Tom"
courses
noC 2
Cr4
CourseRecord
title "EECS281"
marks 77
Cr5
CourseRecord
title "EECS280"
marks 61


Cv2
s1.courses[1]
sms.students[0].courses[1]
@t1 = @t2 ∴ s1 and sms.students[0] point to the same object
∴ s1.courses[1] and sms.students[0].courses[1] point to same object
SMS
students
nos
2
sms
array
sms.studes. X
sms.studes[1].courses[1]
Java:
Diagram:
Student
name "Jim"
courses
noc 3
s1
Student
name "Tom"
courses
noc 2
s2
0 1 2 3 4
CourseRecord
title "EECS100"
marks 72
Cv1
CourseRecord
title "EECS200"
marks 65
Cv2
CourseRecord
title "EECS201"
marks 81
Cv3
s1.courses[2]
CourseRecord
title "EECS201"
marks 77
Cv4
CourseRecord
title "EECS200"
marks 61
Cv5
sms.students[1].courses[1]
How many expressions store the address of this object?


sms.addCourseRecord("Tom", crb);
index of Student ?!
SMS
Students
nos
2
0 1 2 3 4 5
sms.students[i].courses[2]=crb;
sms.students[i].noc =
sms.students[i].noc + 1;
s2.addCourseRecord(crb)
s1
Student
name "Jim"
courses
noc 3
0 1 2 3 4
cr1
CourseRecord
title "EECS101"
marks 72
cr2
CourseRecord
title "EECS201"
marks 65
cr3
CourseRecord
title "EECS221"
marks 81
s2
Student
name "Tom"
courses
noc 2 3
0 1 2 3 4
cr4
CourseRecord
title "EECS101"
marks 77
cr5
CourseRecord
title "EECS201"
marks 61
crb
CourseRecord
title "EECS221"
marks 93


sms.getMarks("Tom","EECS2030");
sms.setMarks("Tom","EECS2030",64);
sms
SMS
students
nos
2
0 1 2 3 4 5
s1
Student
name
"Jim"
courses
noc
3
0 1 2 3 4
cr1
CourseRecord
title
"EECS1011"
marks
72
cr2
CourseRecord
title
"EECS2030"
marks
65
cr3
CourseRecord
title
"EECS3311"
marks
81
s2
Student
name
"Tom"
courses
noc
3
s2.getMarks("EECS2030");
s2.setMarks("EECS2030",64);
0 1 2 3 4
cr4
CourseRecord
title
"EECS1011"
marks
77
cr5
CourseRecord
title
"EECS2030"
marks
~~64~~
64
cr6
CourseRecord
title
"EECS3311"
marks
93


7.1b → sms.students[0].getGPA()
7.5 → sms.students[i].getGPA()
sms.getGPA()
sum: 14.66
avg: 7.33
sms
SMS
Students
nos
2
0 1 2 3 4 5
s1.getGPA()
sum: 21.5
GPA: 7.16
Student
name "Jim"
courses
noc 3
s1
0 1 2 3 4
cr1
CourseRecord
title "EECS100"
marks 72
B
7.0
cr2
CourseRecord
title "EECS200"
marks 65
C+
6.5
cr3
CourseRecord
title "EECS211"
marks 81
A
8.0
s2.getGPA()
sum: 22.5
GPA: 7.5
Student
name "Tom"
courses
noc 3
s2
0 1 2 3 4
cr4
CourseRecord
title "EECS100"
marks 77
B+
7.5
cr5
CourseRecord
title "EECS200"
marks 64
C
6.0
cr6
CourseRecord
title "EECS211"
marks 93
A+
9.0