Monday March 26

Lecture 11

Schedule After Strike

Week #1	Week # 2	Week #3	Revised Exam Date
LAB TEST 2	Grading of Lab 5 & 6	LAB TEST 3	EXAM
(10 %)	(3% 3%)	(10%)	(52 %)

END OF STRIKE

What's Expected during Strike

- LAB TEST 2 prep.
- LAB 5 Background Study
  - ~ PDF
  - ~ Tutorial Videos
- LAB 5 & 6
- Lectures

To Be Released

- LAB TEST 3 guide
- Exam Guide

Tutoring Hours

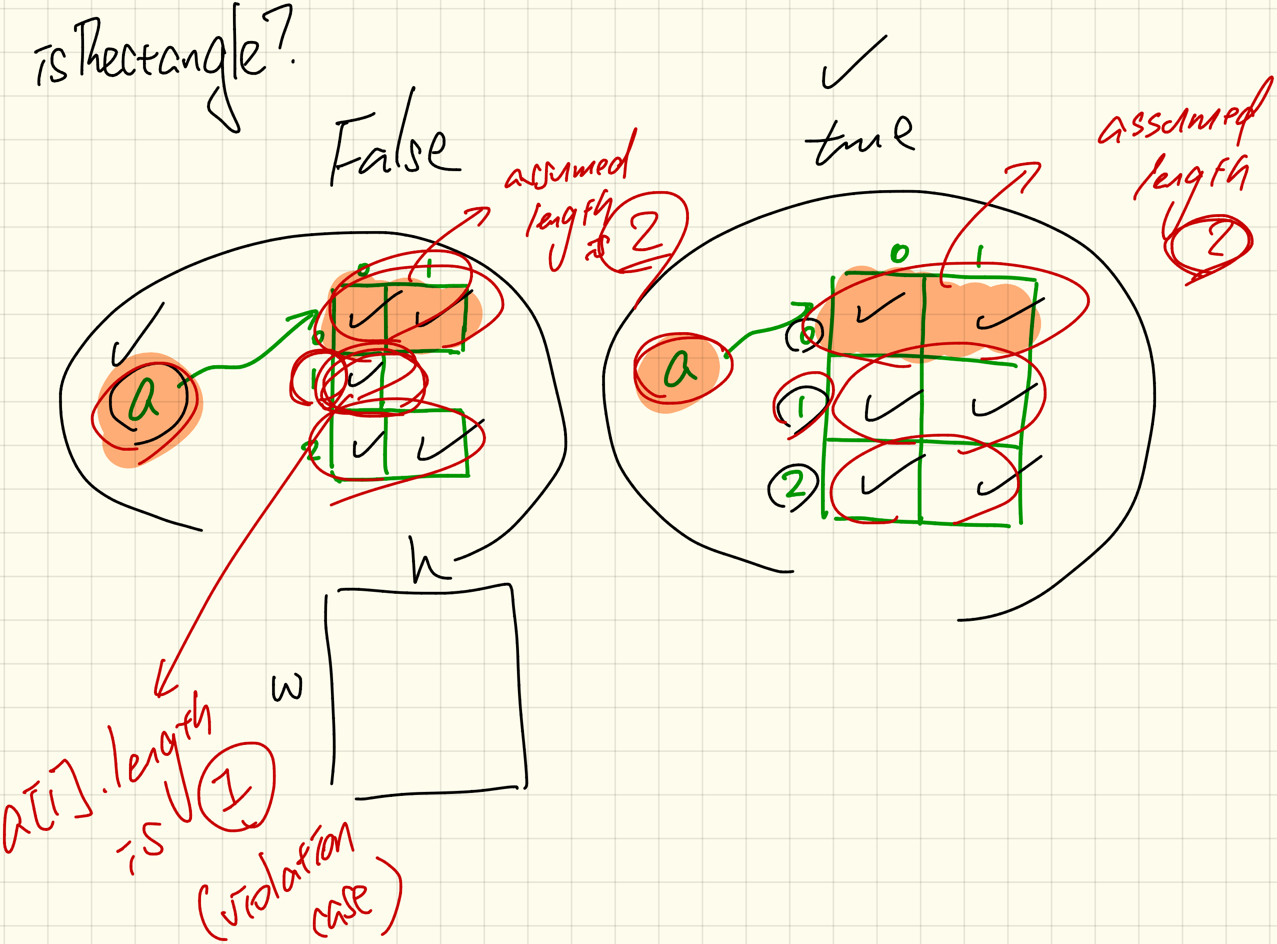
Tuesdays
Wednesdays
Thursdays } 14:30 ~ 15:30
Prism Lab (LAS 1006)

Or, Zoom appointments!

WSC Drop-In Sessions (no TA supervision)

Tuesdays	14:00 ~ 16:00
Wednesdays	17:00 ~ 19:00
Fridays	11:00 ~ 13:00

isRectangle?

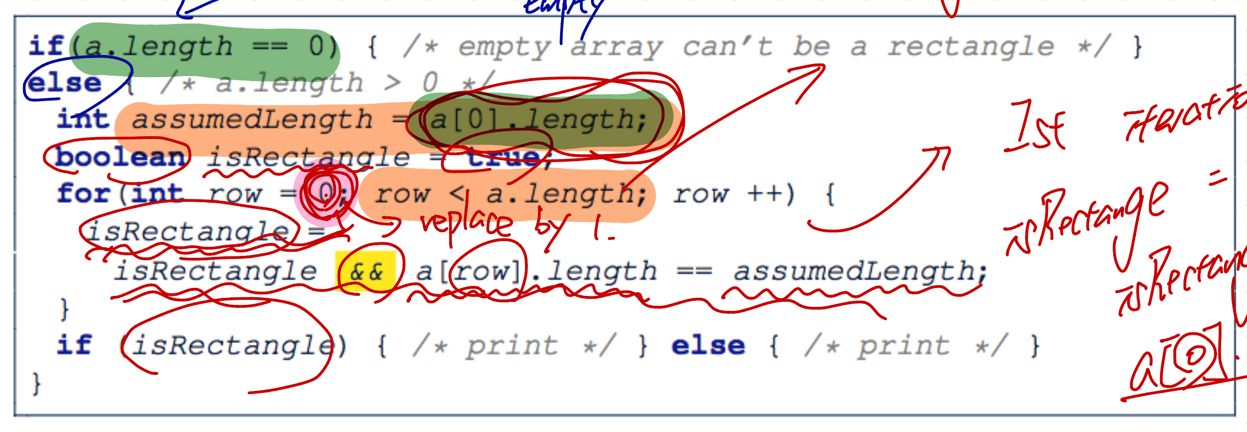


Example 6: isRectangle?

a = { }

assume a is not empty

isRectangle && row < a.length

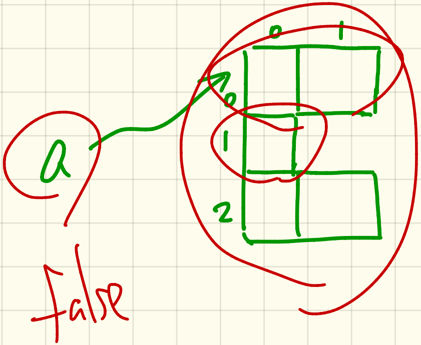


```
if(a.length == 0) { /* empty array can't be a rectangle */ }
else { /* a.length > 0 */
  int assumedLength = a[0].length;
  boolean isRectangle = true;
  for(int row = 0; row < a.length; row ++) {
    isRectangle =
      isRectangle && a[row].length == assumedLength;
  }
  if (isRectangle) { /* print */ } else { /* print */ }
}
```

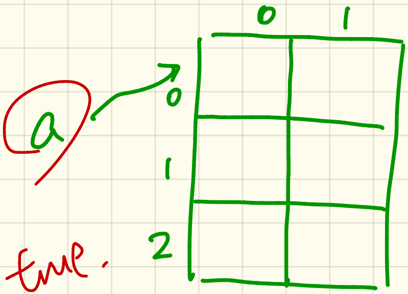
replace by 1.

1st iteration: (row == 0)

isRectange =
isRectangle &&
a[0].length ==
assumedLength;

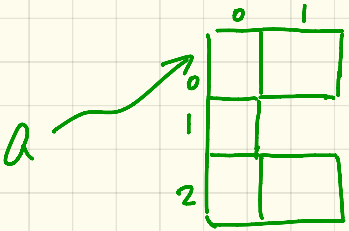


false

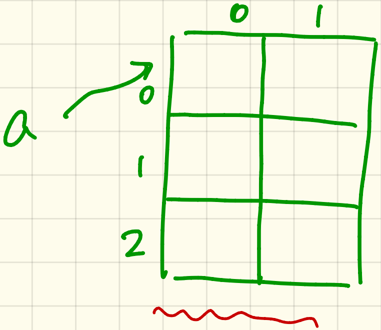


true.

isSquare?

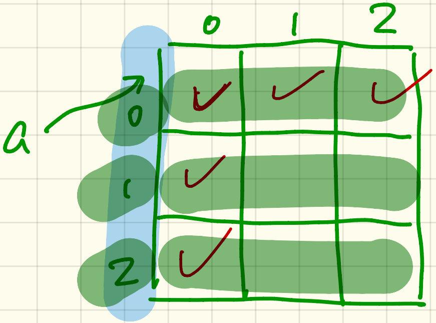


false



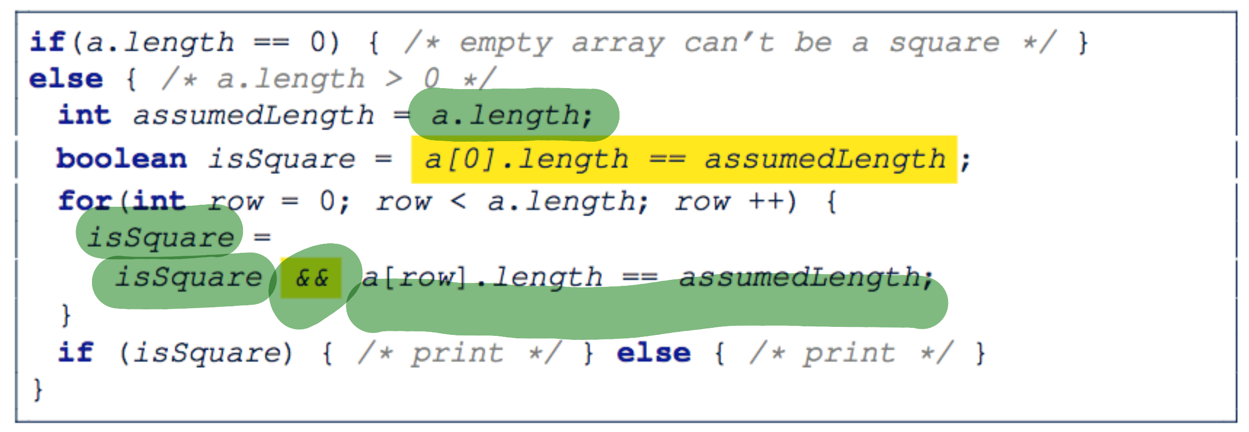
false

a.length == 3

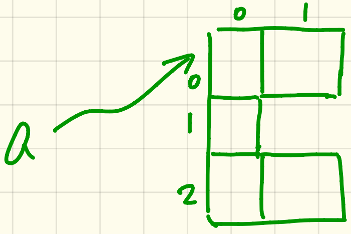


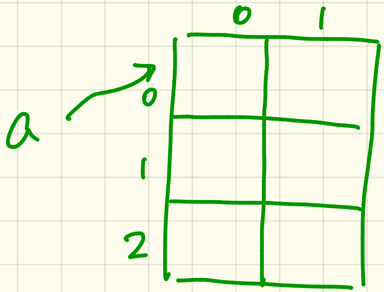
true

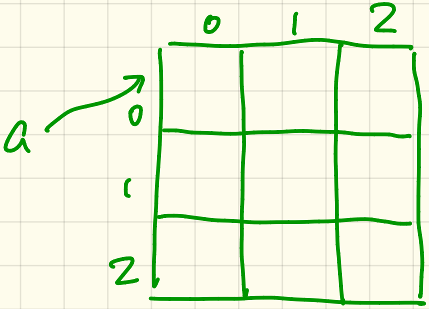
Example 7: isSquare



```
if(a.length == 0) { /* empty array can’t be a square */ }
else { /* a.length > 0 */
  int assumedLength = a.length;
  boolean isSquare = a[0].length == assumedLength;
  for(int row = 0; row < a.length; row ++) {
    isSquare =
      isSquare && a[row].length == assumedLength;
  }
  if (isSquare) { /* print */ } else { /* print */ }
}
```

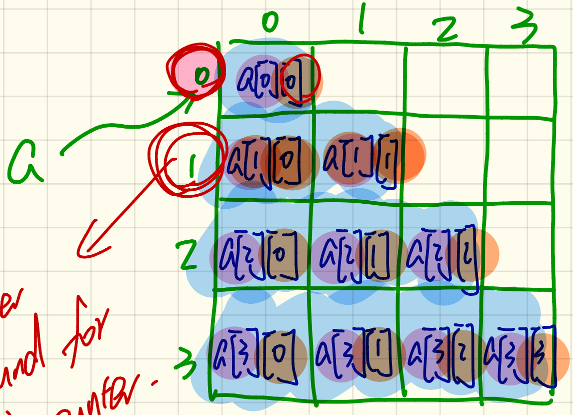






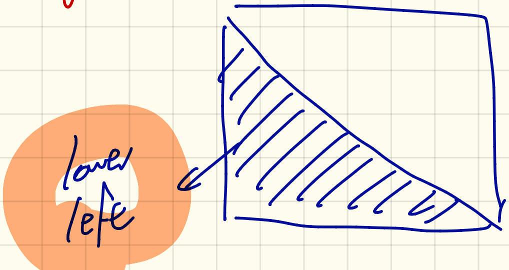
Print lower left?

print upper left?

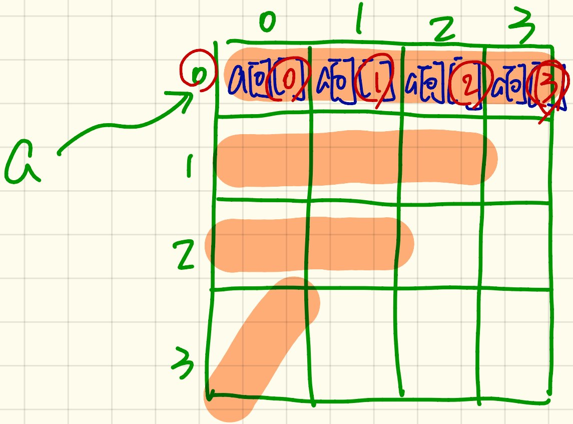


	0	1	2	3
0	a[0][0]			
1	a[1][0]	a[1][1]		
2	a[2][0]	a[2][1]	a[2][2]	
3	a[3][0]	a[3][1]	a[3][2]	a[3][3]

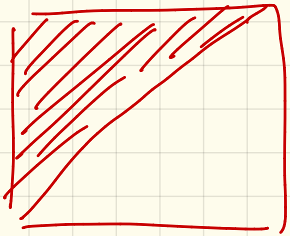
upper bound for col counter.



$0 \ldots \; a[\underset{\text{row}}{0}].\text{length} - \underset{0}{\text{row}}$

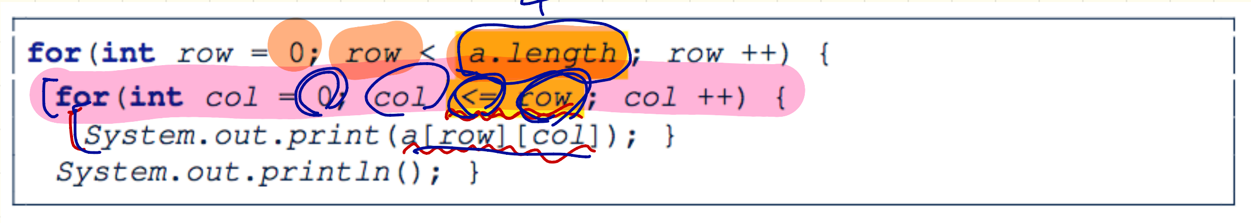


	0	1	2	3
0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
1				
2				
3				

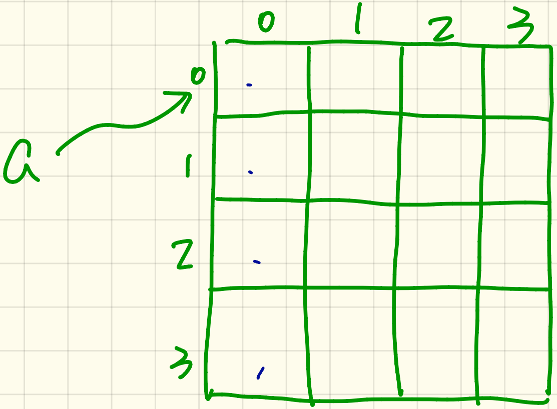


Example 8: Lower Left

4

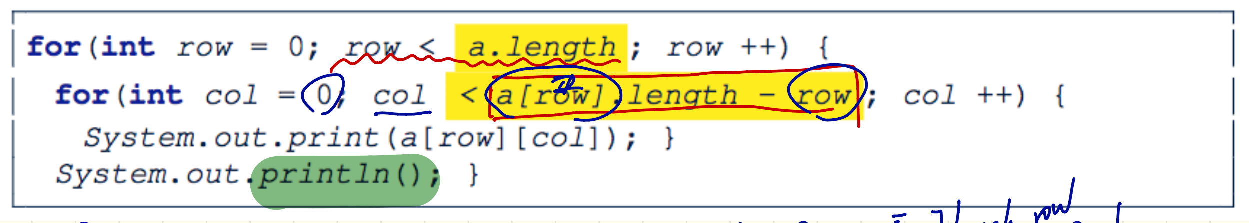


```
for(int row = 0; row < a.length; row ++) {
  for(int col = 0; col <= row; col ++) {
    System.out.print(a[row][col]); }
  System.out.println(); }
```



row	col
0	0
1	0 1
2	0 1 2
3	0 1 2 3

Example 9: Upper Left



```
for(int row = 0; row < a.length ; row ++) {
  for(int col = 0; col < a[row].length - row; col ++) {
    System.out.print(a[row][col]); }
  System.out.println(); }
```

0 Output

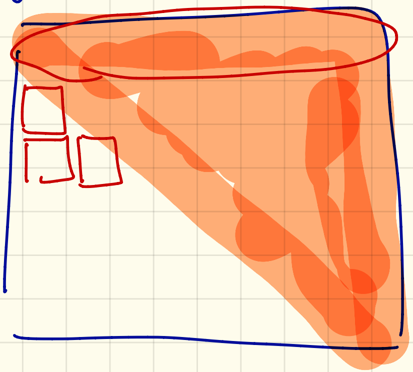
```
4 3 8 12
1 2 9
3 6
4
```

a

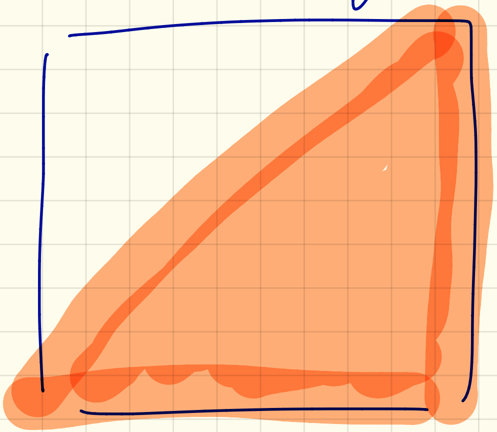
	0	1	2	3
0	4	3	8	12
1	1	2	9	13
2	3	6	7	14
3	4	10	11	15

row	a[row].length-row	col
0	4-0	0 1 2 3
1	4-1	0 1 2
2	4-2	0 1
3	4-3	0

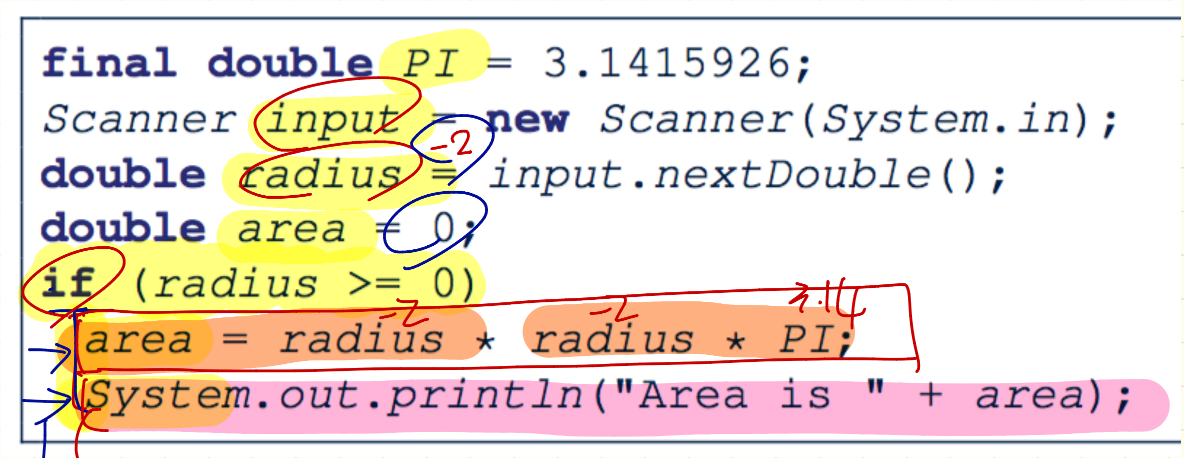
Upper Right



Lower Right



Selections: Missing Brackets



```
final double PI = 3.1415926;
Scanner input = new Scanner(System.in);
double radius = input.nextDouble();
double area = 0;
if (radius >= 0)
  area = radius * radius * PI;
  System.out.println("Area is " + area);
```

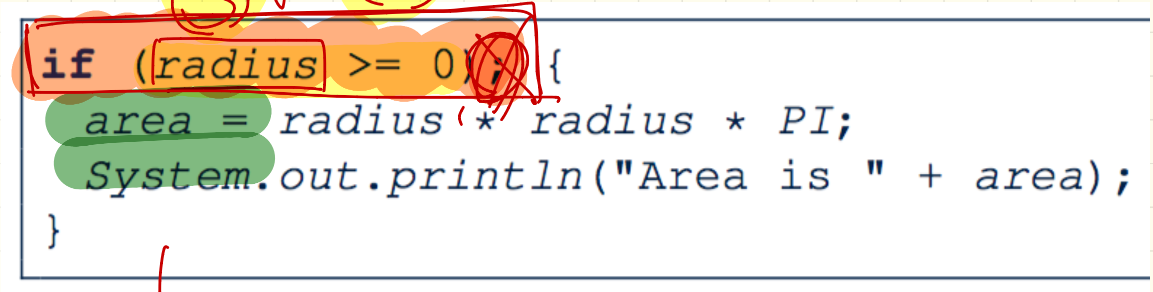
≡

-2

```
if (radius >= 0) {
    area = - -
}
println ( Area - - )
```

Selections: Misplaced Semicolon

3*3*PI ← 3 vs. -2 → -2*-2* PI



```
if (radius >= 0); {
  area = radius * radius * PI;
  System.out.println("Area is " + area);
}
```

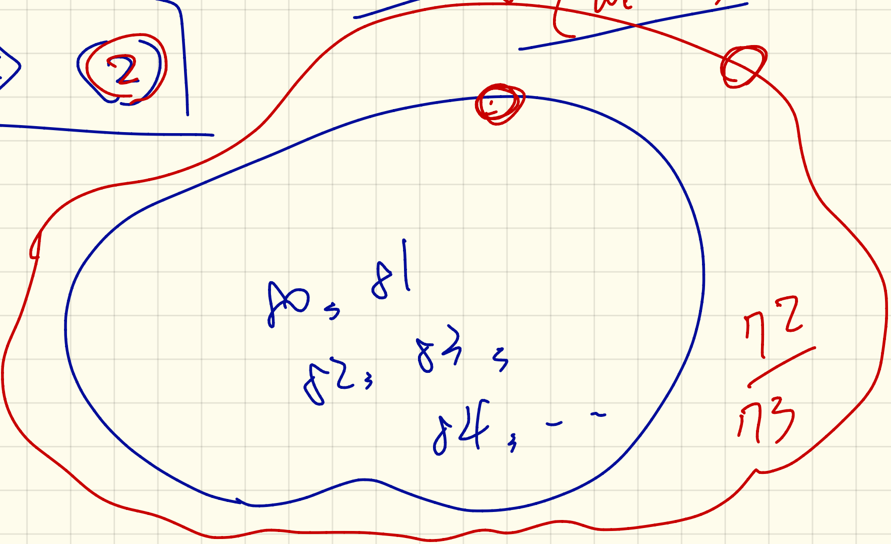
```
if ( radius >= 0) { }
area = . - - ;
println( area)
```

Overlapping Boolean Conditions.

① marks >= 80 → more specific (stronger)

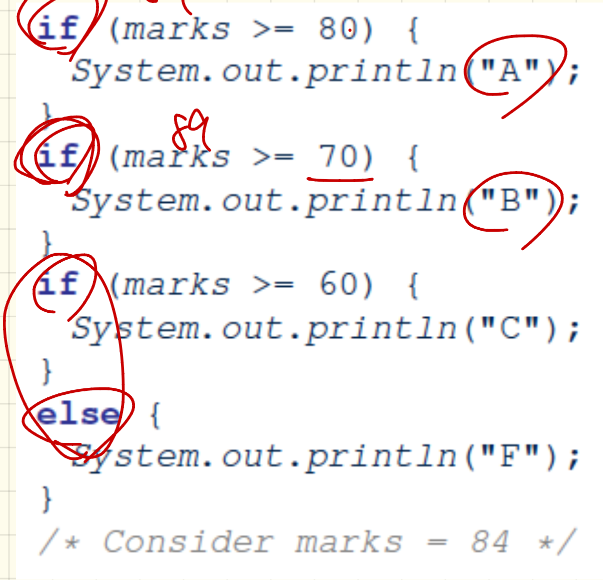
81 ② 75 marks >= 70 → more general (weaker)

① ⇒ ②



Common Errors of Selections:

Independent if-statements with overlapping conditions.



```
if (marks >= 80) {
  System.out.println("A");
}
if (marks >= 70) {
  System.out.println("B");
}
if (marks >= 60) {
  System.out.println("C");
}
else {
  System.out.println("F");
}
/* Consider marks = 84 */
```

marks : 89

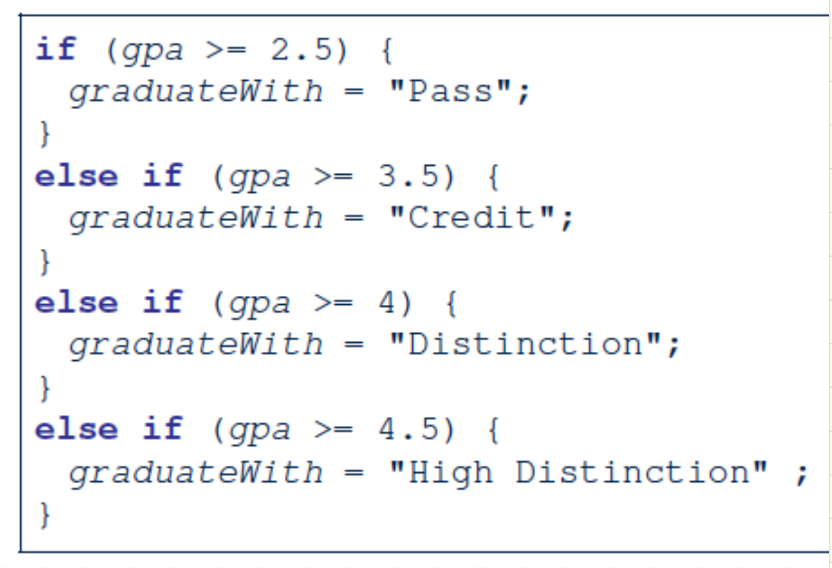
A

B

C

Common Errors of Selections:

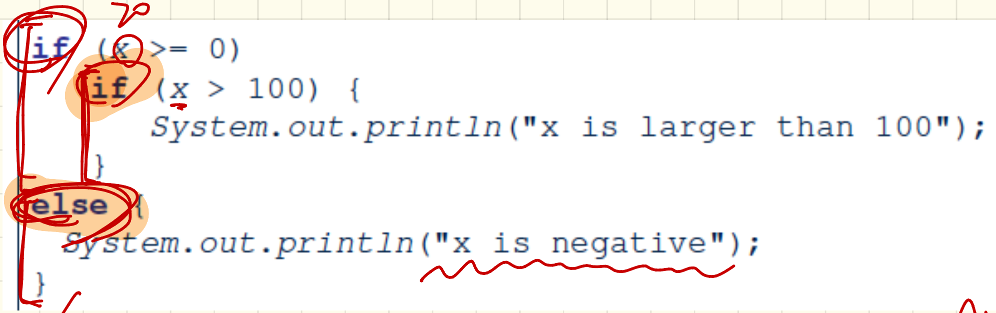
If-conditions sorted the wrong way



```
if (gpa >= 2.5) {
  graduateWith = "Pass";
}
else if (gpa >= 3.5) {
  graduateWith = "Credit";
}
else if (gpa >= 4) {
  graduateWith = "Distinction";
}
else if (gpa >= 4.5) {
  graduateWith = "High Distinction" ;
}
```

Common Errors of Selections:

Ambiguous else-statement



20

```
if (x >= 0)
  if (x > 100) {
     System.out.println("x is larger than 100");
  }
else {
  System.out.println("x is negative");
}
```

What if x is 20

```
if (x >= 0) {
    if (x > 100) {
                  ____________
    }
    else {
    ____
    }
}
```

parsing rule

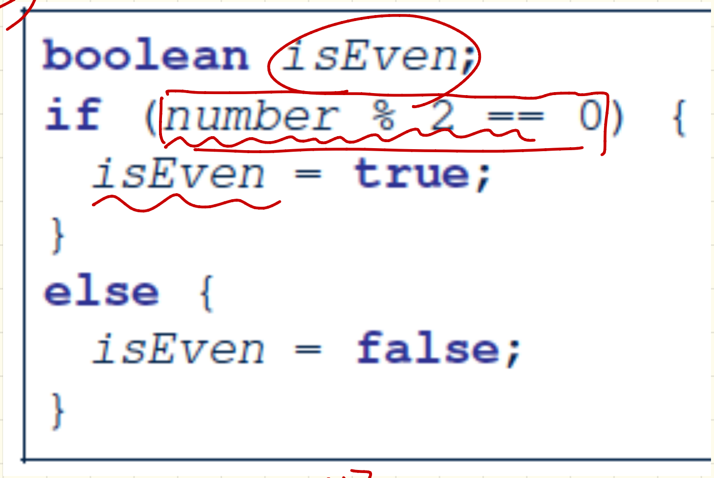
Ex

```
if (x >= 0) {
    if (x > 100) {
    ____________
    }
}
else {
    println( -  — neg.);
}
```

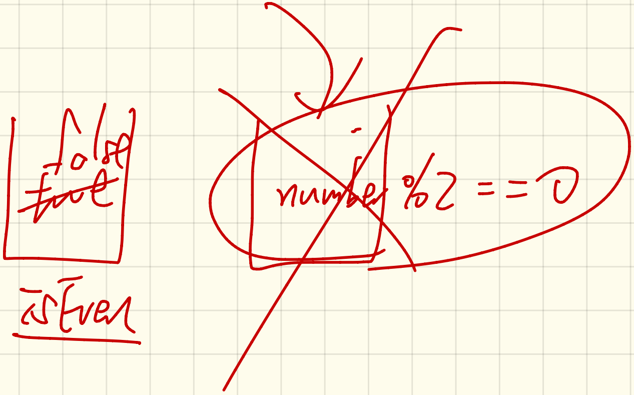
Common Pitfalls of Selections:

Redundant if

v1



```
boolean isEven;
if (number % 2 == 0) {
  isEven = true;
}
else {
  isEven = false;
}
```



v2

isEven = number % 2 == 0 ;

V1

```
boolean
  isOdd ;
   3
if( number % 2 == 0){
      isOdd = false;
}
else {
       isOdd = true ;
 }
```

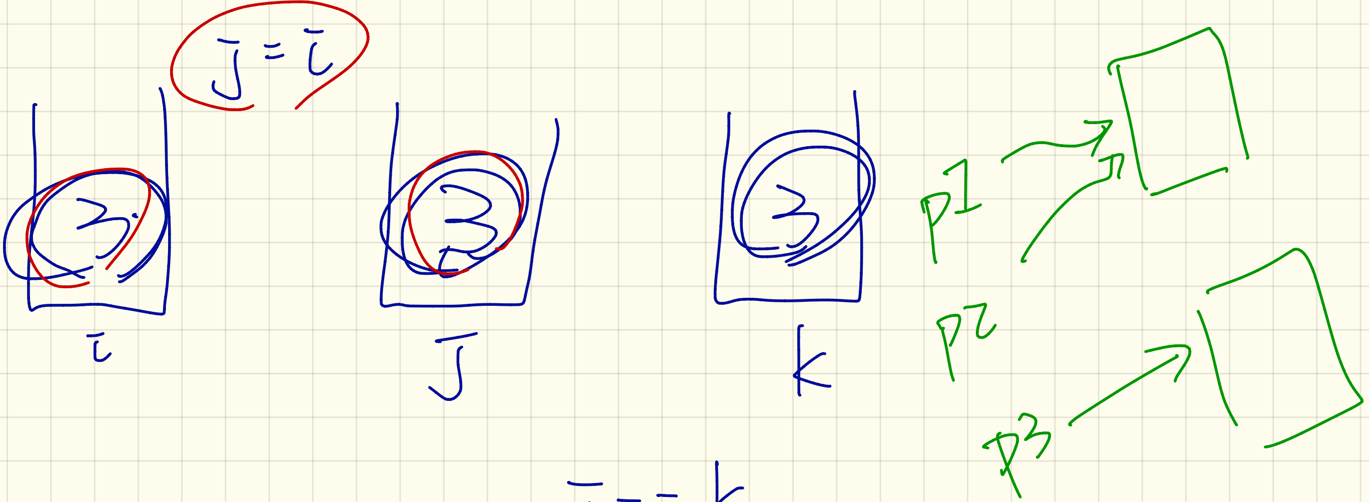
```
                     3
X isOdd = number % 2==0;
Fix1: isOdd = num % 2==1
Fix2: isOdd = !(num % 2 ==0)
Fix3: isOdd = num %2 != 0
```

Primitive

int
short
long
float
double
boolean
char

Reference

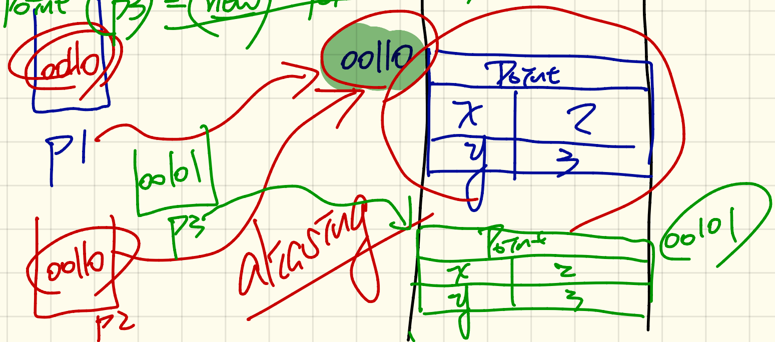
String
Point
Person



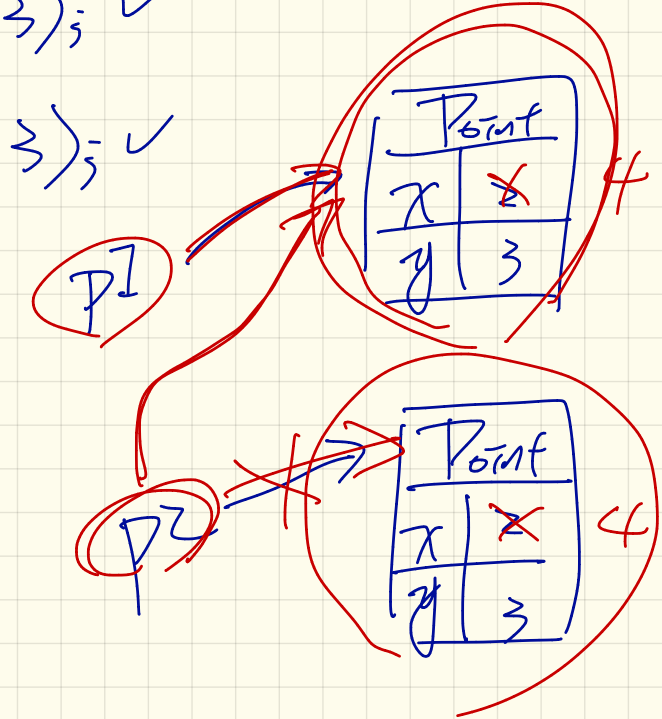
j == k

```
class Point {
  double x;
  double y;
  Point ( double x, double y){
    this.x = x;   this.y=y;
  }
}
```

```
Point p1 = new Point(2,3);
Point p2 = p1;
Point p3 = new Point(2,3);
```



```
Point p1 = new Point(2, 3); ✓
Point p2 = new Point(2, 3); ✓
println( p1 == p2 );
println( p1.x );   /* 2 */
p2. x = 4;
println(p1. x);    /* 2 */
p2 = p1;  *
p2.x = 4;
println(p1.x); ✓   /* 4 */
```



p1
p2
Point
x
~~2~~
4
y
3
Point
x
~~2~~
4
y
3