

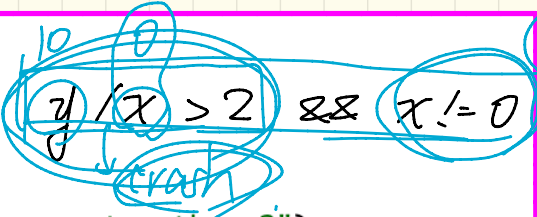
Wednesday January 30  
Lecture 8

- Lab 2 released
- Quiz 2 guide released
- Lab Sessions 09/10 special agenda  
to be sent tomorrow

# Short-Circuit Evaluation: $\&\&$

Left Operand op1	Right Operand op2	op1 $\&\&$ op2
true	true	true
true	false	false
false	true	false
false	false	false

```
System.out.println("Enter x:");
int x = input.nextInt();
System.out.println("Enter y:");
int y = input.nextInt();
if (x != 0 && y / x > 2) {
    System.out.println("y / x is greater than 2");
}
else { /* !(x != 0 && y / x > 2) == (x == 0 || y / x <= 2) */
    if (x == 0) {
        System.out.println("Error: Division by Zero");
    }
    else {
        System.out.println("y / x is not greater than 2");
    }
}
```



Test Case:  
x = 0  
y = 10

Test Case:  
x = 5  
y = 10

# Short-Circuit Evaluation: ||

Left Operand op1	Right Operand op2	op1    op2
false	false	false
→ true	false	true
false	true	true
→ true	true	true

$x \neq 0 \vee y/x > 2$   
 $\neg(x=0) \vee y/x > 2$

$y/x > 2 \vee x \neq 0$   
 $(10/0) > 2 \rightarrow \text{crash}$

```

System.out.println("Enter x:");
int x = input.nextInt();
System.out.println("Enter y:");
int y = input.nextInt();
→ if(x == 0 || y / x > 2) {
    if(x == 0) {
        System.out.println("Error: Division by Zero");
    }
    else {
        System.out.println("y / x is greater than 2");
    }
}
else { /* !(x == 0 || y / x > 2) == (x != 0 && y / x <= 2) */
    System.out.println("y / x is not greater than 2");
}
    
```

Test Case:  
 $x = 0$   
 $y = 10$

Test Case:  
 $x = 5$   
 $y = 10$

# Short-Circuit Evaluation: Common Error

Test Case :

$x = 0$

$y = 10$

crash when  
 $x == 0$

Short-Circuit Evaluation is not exploited: crash when  $x == 0$

```
if (y / x > 2 && x != 0) {  
    /* do something */  
}  
else {  
    /* print error */ }
```

crash when  $x == 0$

Short-Circuit Evaluation is not exploited: crash when  $x == 0$

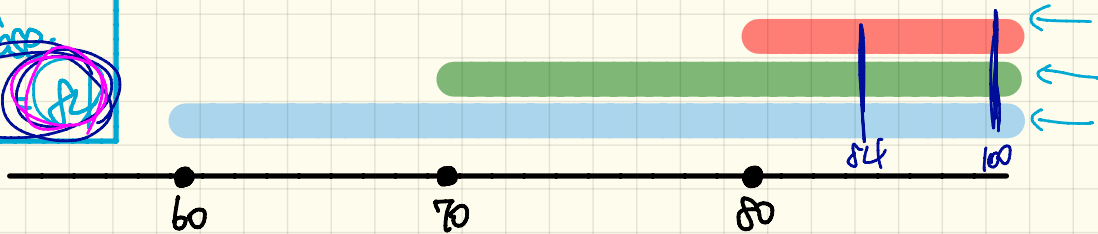
```
if (y / x <= 2 || x == 0) {  
    /* print error */  
}  
else {  
    /* do something */ }
```

# Common Error: Overlapping Conditions in Multiple If-Statements

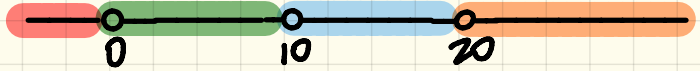
```
if (marks >= 80) {  
    System.out.println("A");  
}  
if (marks >= 70) {  
    System.out.println("B");  
}  
if (marks >= 60) {  
    System.out.println("C");  
}  
else {  
    System.out.println("F");  
}
```

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

Test Case  
marks = 84



# Exercise: Overlapping Conditions



Does this program always print exactly one line?

```
if (x < 0) { println("x < 0"); }
if (0 <= x && x < 10) { println("0 <= x < 10"); }
if (10 <= x && x < 20) { println("10 <= x < 20"); }
if (x >= 20) { println("x >= 20"); }
```

multiple if's non-overlapping

Does this program always print exactly one line?

```
if (x < 0) { println("x < 0"); }
else if (0 <= x && x < 10) { println("0 <= x < 10"); }
else if (10 <= x && x < 20) { println("10 <= x < 20"); }
else if (x >= 20) { println("x >= 20"); }
```

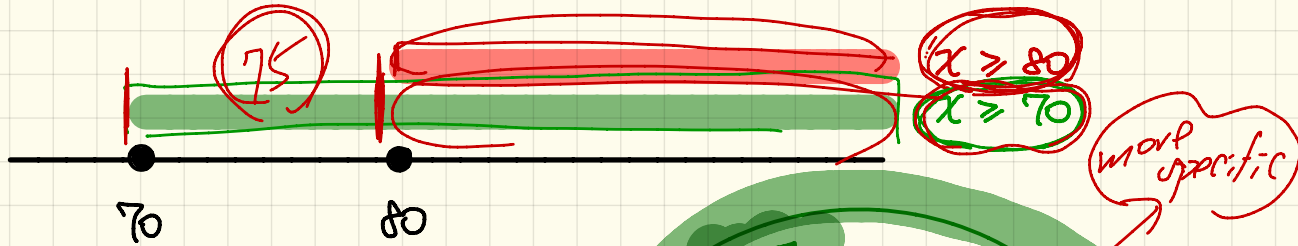
single if's non-overlapping

This simplified version is equivalent:

```
if (x < 0) { println("x < 0"); }
else if (x < 10) { println("0 <= x < 10"); }
else if (x < 20) { println("10 <= x < 20"); }
else { println("x >= 20"); }
```

$!(x < 0) = x >= 0$

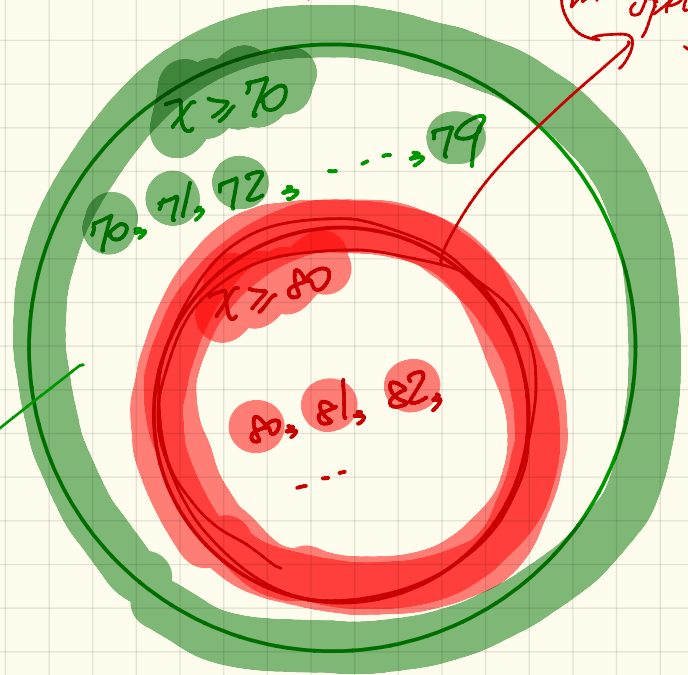
# Overlapping Conditions: General vs. Specific



$x \geq 70$  is more general.

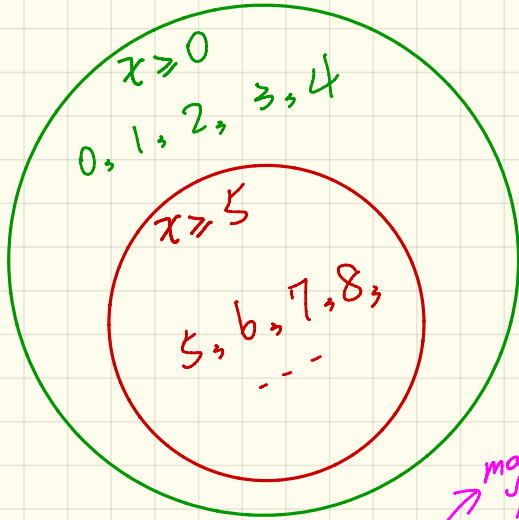
$x \geq 80$  is more specific.

more general

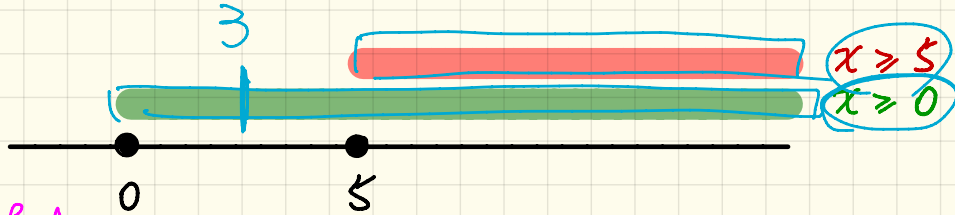




# Overlapping Conditions in a Single If-Statement



$x \geq 0$  IS more general.  
 $x \geq 5$  IS more specific.



✓ If we have a single if statement, then having this order

```
if (x >= 5) { System.out.println("x >= 5"); }  
else if (x >= 0) { System.out.println("x >= 0"); }
```

is different from having this order

```
if (x >= 0) { System.out.println("x >= 0"); }  
else if (x >= 5) { System.out.println("x >= 5"); }
```

Test Case:

$x = 5$

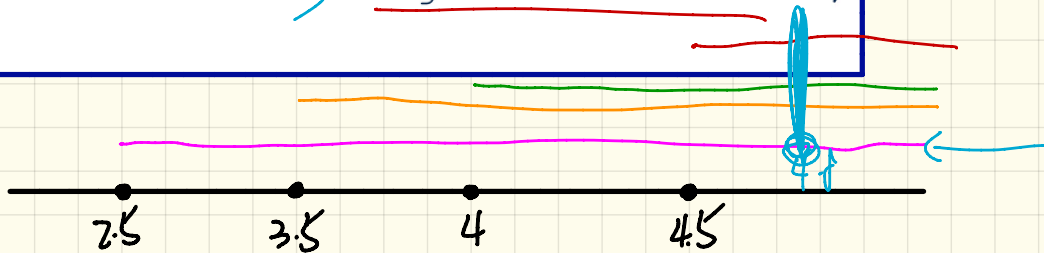
more specific

more general

# Common Error: General Condition comes before Specific Condition

```
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";  
}
```

Test Case:  
gpa = 4.8



## Common Error: Missing Braces

Confusingly, braces can be omitted if the block contains a **single** statement.

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

Your program will *misbehave* when a block is supposed to execute **multiple statements**, but you forget to enclose them within braces.

```
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);
```

Test Case:  
radius = -3

## Common Error: Mispaced Semicolon

Test Case:

radius = -4

Semicolon (;) in Java marks *the end of a statement* (e.g., assignment, if statement).

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

Annotations:  
- A blue arrow points to the opening curly brace of the if statement.  
- A pink circle highlights the semicolon after the if condition, with a pink arrow pointing to it from the text "start of something unconditional".  
- A green circle highlights the closing curly brace of the if statement.  
- A pink arrow points to the opening curly brace of the if statement.  
- A pink arrow points to the assignment statement.  
- A pink arrow points to the println statement.  
- A green circle highlights the closing curly brace of the if statement.

This program will calculate and output the area even when the input radius is *negative*, why? Fix?

# Common Error: Variable Not Properly Reassigned

= "" "Fail!"

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

Test Case:

gpa: 1.5

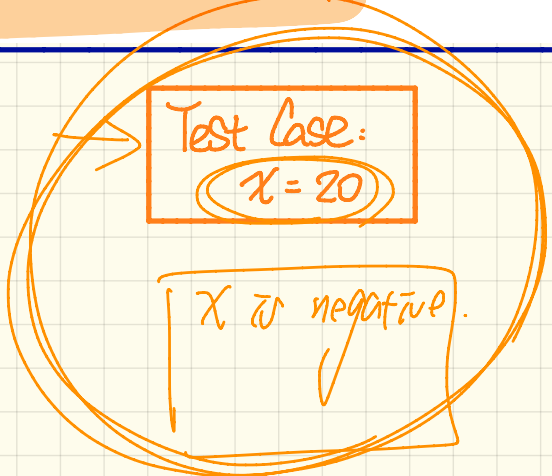
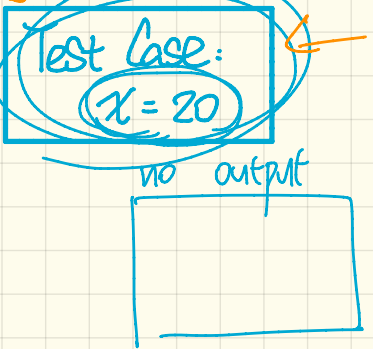
else {

else {

gW = "Fail"

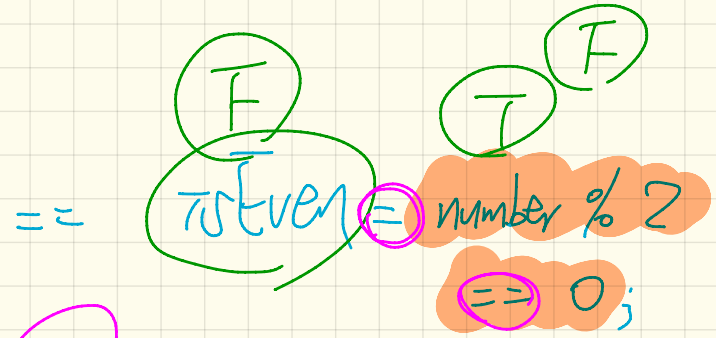
# Common Error: Ambiguous "else"

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



# Common Pitfall

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



```
if (isEven == false) {  
    System.out.println("Odd Number");  
}  
else {  
    System.out.println("Even Number");  
}
```



$T$

$!isEven$   
 $F$   $T$