## COMP I I 0/L Lecture I I

Mahdi Ebrahimi

Slides adapted from Dr. Kyle Dewey

# Outline

- @Test**vs.**assertEquals
- Boolean operations
  - & & • | |
  - Complex if cond
- Complex if conditions

# @Test vs. assertEquals

# @Test vs. assertEquals

- @Test defines a test
- assertEquals checks a condition
- Can have a @Test containing no assertEquals
  - Test always passes
- Can have multiple assertEquals per @Test
  - Test passes if all assertEquals are ok

## **Example:** MultiAssert.java MultiAssertTest.java

You're already familiar with operations returning boolean

You're already familiar with operations returning boolean

#### 3 < 6

You're already familiar with operations returning boolean

You're already familiar with operations returning boolean

Can chain boolean expressions with AND (&&). Semantics: only true if both sides are true.

Can chain boolean expressions with AND (&&). Semantics: only true if both sides are true.

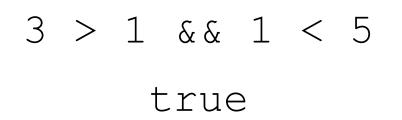
#### 3 > 1 && 1 < 5

Can chain boolean expressions with AND (&&). Semantics: only true if both sides are true.

#### 3 > 1 && 1 < 5

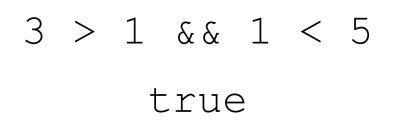
true

Can chain boolean expressions with AND (&&). Semantics: only true if both sides are true.



#### 1 > 3 && 1 < 5

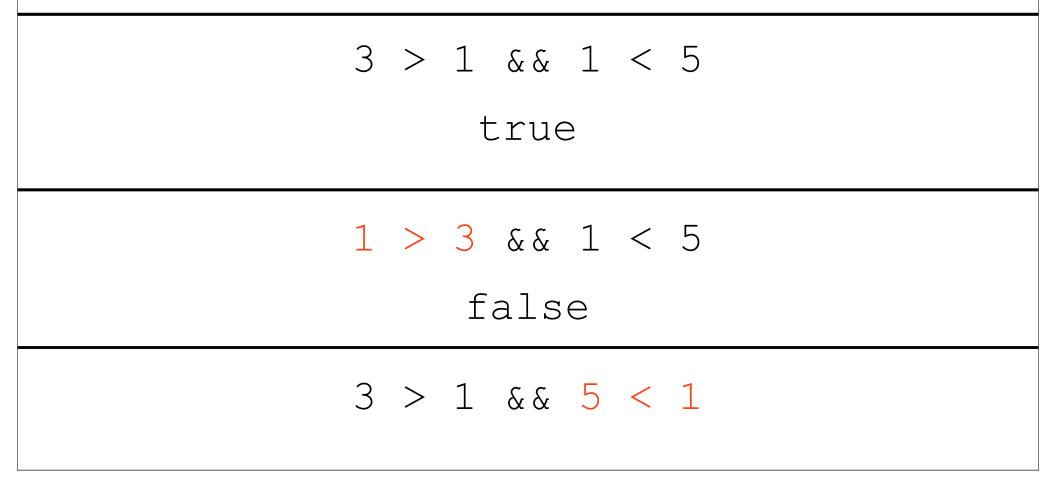
Can chain boolean expressions with AND (&&). Semantics: only true if both sides are true.

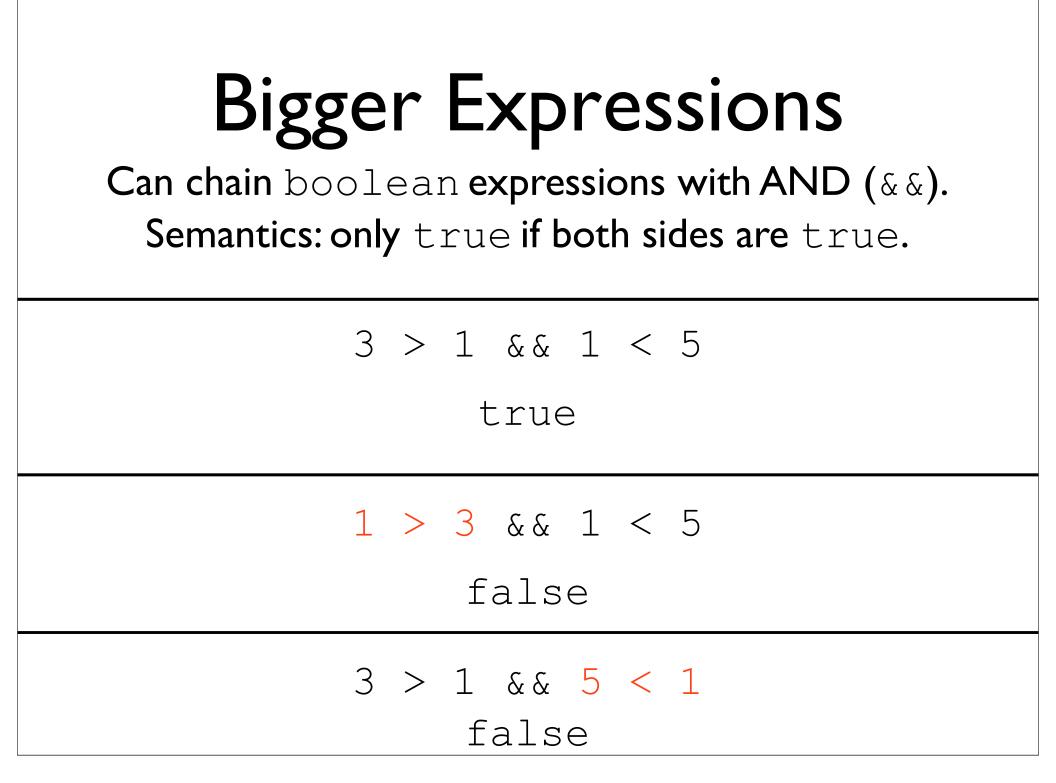


1 > 3 && 1 < 5 false



Semantics: only true if both sides are true.





## Truth Table

Truth tables show the result of combining any two expression boolean expressions using the **AND** operator and the **OR** operator (or the **NOT** operator). You should memorize/learn these values.

condition 1 (e.g., X)	condition 2 (e.g., Y)	X AND Y (X & & Y )
false	false	false
false	true	false
true	false	false
true	true	true

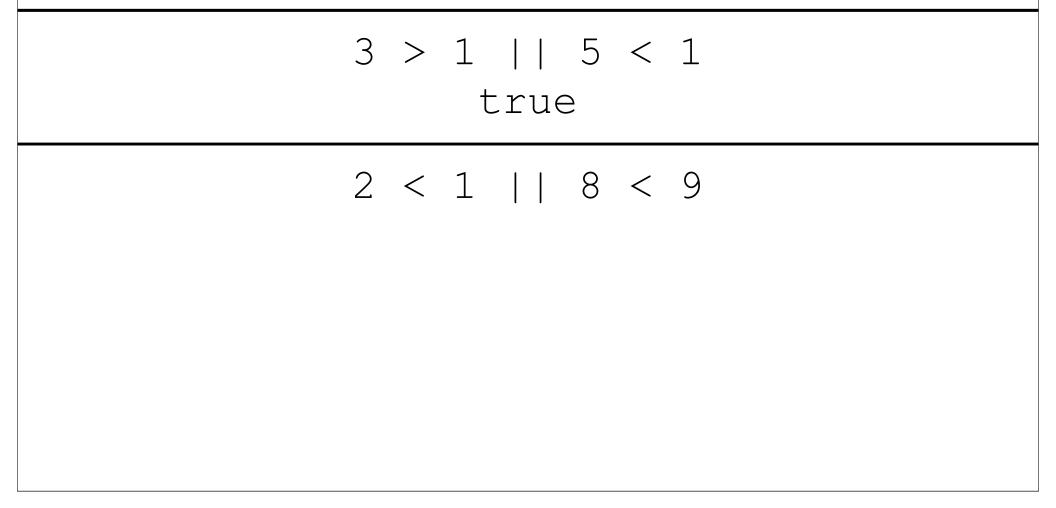
# **Example:** And.java

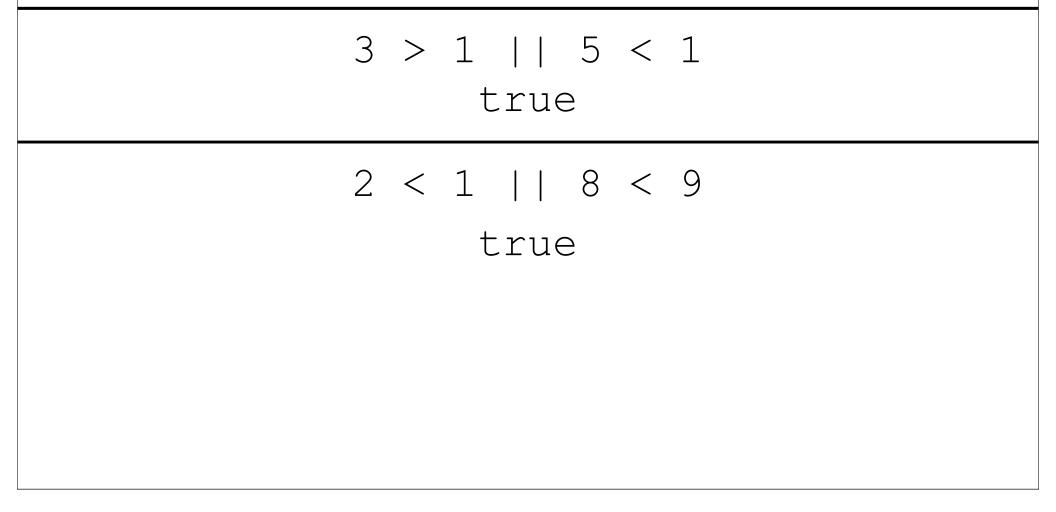
boolean expressions can also be combined with OR (||) Semantics: true if either side is true.

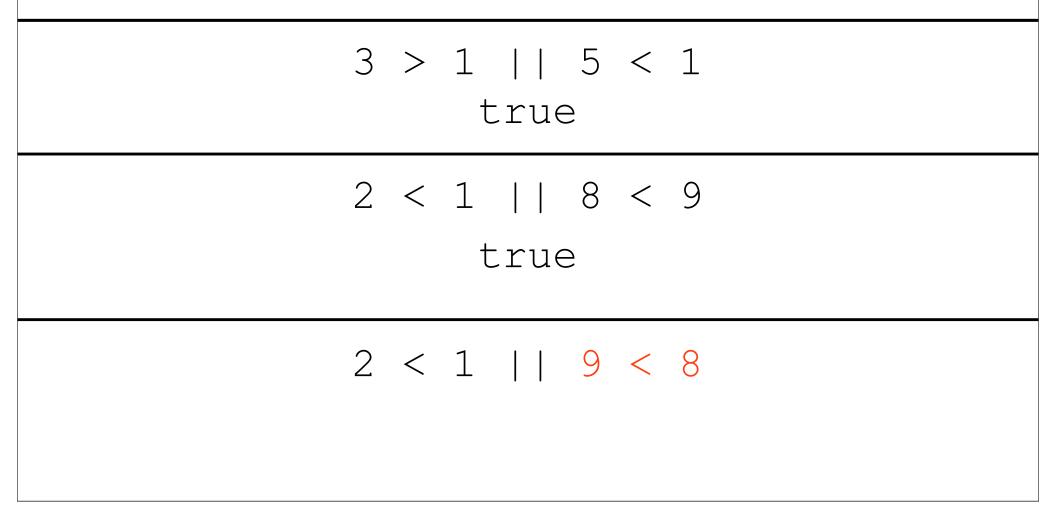
3 > 1 || 5 < 1

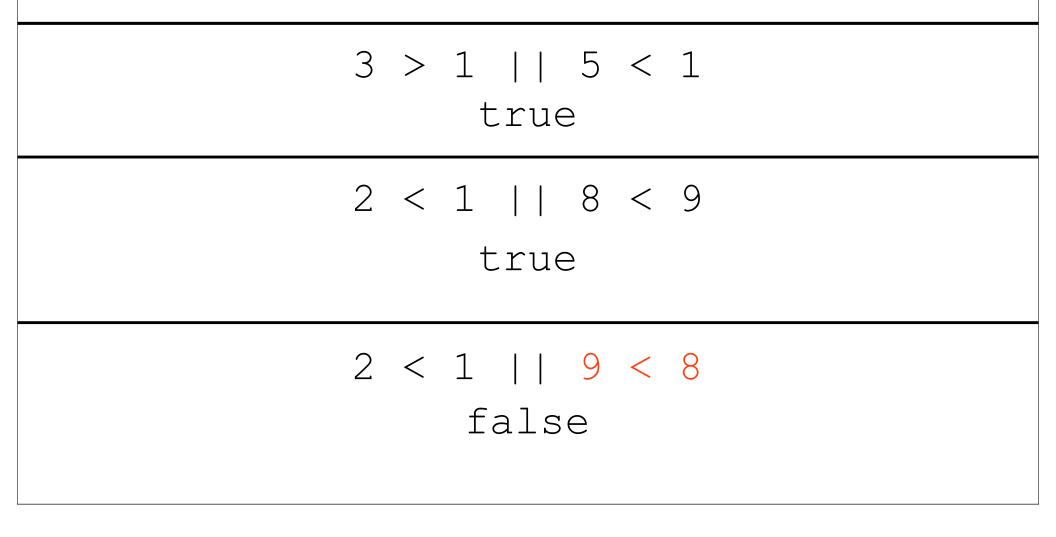
boolean expressions can also be combined with OR (||) Semantics: true if either side is true.

#### 3 > 1 || 5 < 1 true









## Truth Table

Truth tables show the result of combining any two expression boolean expressions using the **AND** operator and the **OR** operator (or the **NOT** operator). You should memorize/learn these values.

condition 1 (e.g., X)	condition 2 (e.g., Y)	X OR Y ( X    Y )
false	false	false
false	true	true
true	false	true
true	true	true

# **Example:** Or.java

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

! (1 < 2)

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

!(1 < 2) false

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

!(1 < 2) false

!(1 > 7)

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

!(1 < 2) false

!(1 > 7)

true

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

!(1 < 2) false

true

! (1 < 2 && 1 > 3)

Can negate a boolean expression with not (!).
Semantics: !true == false and !false == true.

!(1 < 2) false

true

!(1 < 2 && 1 > 3) true

## Truth Table

Truth tables show the result of combining any two expression boolean expressions using the **AND** operator and the **OR** operator (or the **NOT** operator). You should memorize/learn these values.

condition 1 (e.g., X)	NOT X ( !X )
false	true
true	false

# **Example:** Not.java

## Truth Table

Truth tables show the result of combining any two expression boolean expressions using the **AND** operator and the **OR** operator (or the **NOT** operator). You should memorize/learn these values.

condition 1	condition 2	NOT X	X AND Y	X OR Y
(e.g., X)	(e.g., Y)	( !X )	( X && Y )	( X    Y )
false	false	true	false	false
false	true	true	false	true
true	false	false	false	true
true	true	false	true	true

#### Putting it Together: ComplexConditional.java

Operators	Description	Associativity	
0	Function Call		
	Member Selection	Left to Right	
I	Logical negation	Right to Left	
*	Multiplication		
/	Division	Left to Right	
%	Modulo		
+ -	Addition / Subtraction	Left to Right	
< <=	Relational Less Than / Less than Equal To		
> >=	Relational Greater / Greater than Equal To	Left to Right	
==	Equality	Left to Right	
!=	Inequality		
&&	Logical AND	Left to Right	
Ш	Logical OR	Left to Right	
=	Assignment Operators	Right to Left	

# Testing with Boolean Operations

Uses of & & and | | usually mean more tests are appropriate

# Testing with Boolean Operations

Uses of & & and | | usually mean more tests are appropriate

if (x == 1 || x == 5) { return 7;

- } else if (x > 7 && x <= 20) {
   return 8;</pre>
- } else {
   return 55;

#### **Testing with Boolean** Operations Uses of && and || usually mean more tests are appropriate Test: x = 1if (x == 1 | | x == 5) { return 7; } else if (x > 7 && x <= 20) { return 8; } else { return 55;

#### **Testing with Boolean** Operations Uses of && and || usually mean more tests are appropriate Test: x = 1 Test: x = 5if (x == 1 | | x == 5) { return 7; } else if (x > 7 && x <= 20) { return 8; } else { return 55;

# Testing with Boolean Operations

Uses of & & and | | usually mean

more tests are appropriate

Test: x = 1 Test: x = 5
if (x == 1 || x == 5) {
 return 7; Test: x = 8
} else if (x > 7 && x <= 20) {
 return 8;
} else {
 return 55;
}</pre>

# Testing with Boolean Operations

Uses of && and  $|\ |$  usually mean

more tests are appropriate

Test: x = 1 Test: x = 5
if (x == 1 || x == 5) {
 return 7; Test: x = 8
} else if (x > 7 && x <= 20) {
 return 8;</pre>

} else {

return 55; **Test**: x = 21

#### Putting it Together: ComplexConditionalTest.java