

Same or Equal?

Comparing Boolean expressions



Two signs at the pool

PAUSE AND TRY

Does each sign say "swim"? Do the two signs always agree?

SIGN 1

Swim unless it is raining or cold.

SIGN 2

Swim if it is not raining or not cold.

Raining?	Cold?	Sign 1	Sign 2
yes	yes		
yes	no		
no	yes		
no	no		

By the end of class you can...

1

Decide whether two Boolean expressions are equivalent, or find one case that proves they aren't.

2

Prove equivalence with a complete truth table.

3

Use De Morgan's laws to rewrite a condition without the outer !.

4

Predict == and != on object references, including aliases and null.

5

Explain why you compare the text of two Strings with equals, not ==.

A large white square with rounded corners, tilted slightly to the right, containing a large, bold, black number '1'.

PART 1

Equivalent expressions

Same value in every case



Two Boolean expressions are **equivalent** if they evaluate to the same value **in all cases**. One case where they differ, a **counterexample**, proves they are not.



Test the value right on the boundary.
The opposite of $<$ is $\geq$, not $>$.

IS "NOT BELOW 32" THE SAME AS "ABOVE 32"?

```
int temp = 32;  
System.out.println(!(temp < 32));  
System.out.println(temp > 32);  
System.out.println(temp >= 32);
```

OUTPUT

```
true  
false  
true
```

A truth table proves it

a	b	a b	!(a b)	!a	!b	!a && !b
true	true	true	false	false	false	false
true	false	true	false	false	true	false
false	true	true	false	true	false	false
false	false	false	true	true	true	true



A **truth table** lists every combination of values. If two columns match in **every** row, the expressions are equivalent. Count the rows: 4 for two variables, 8 for three.

Three matching rows prove nothing



A truth table is a proof only when it is **complete**. Skip a row and you might miss the one case that differs.

a	b	a && b	a == b
true	true	true	true
true	false	false	false
false	true	false	false
false	false	false	true

Rows 1–3 agree.

Row 4 doesn't:

a && b is false,
but a == b is true.
Not equivalent.

Equivalent, or not?

PAUSE AND TRY

For each pair, write "equivalent" or give a counterexample.

	Expression	Expression	Equivalent?
1	<code>!(n > 0)</code>	<code>n <= 0</code>	
2	<code>!(n == 3)</code>	<code>n > 3</code>	
3	<code>!(done late)</code>	<code>!done && !late</code>	
4	<code>!(ready && !tired)</code>	<code>!ready tired</code>	

A large, bold, teal number '2' is centered within a white rounded square that has a slight 3D effect and a shadow.

PART 2

De Morgan's Laws

Negate each part, flip the operator

$!(a \ \&\& \ b)$ is the same as $!a \ || \ !b$

$!(a \ || \ b)$ is the same as $!a \ \&\& \ !b$



De Morgan's laws: push the **!** onto each part, and **flip** the operator: **&&** becomes **||**, and **||** becomes **&&**.

NEGATING A COMPARISON

Comparison	Its opposite
$x < 5$	$x \geq 5$
$x > 5$	$x \leq 5$
$x \leq 5$	$x > 5$
$x \geq 5$	$x < 5$
$x == 5$	$x != 5$

Rewrite it without the !

START

```
!(level > 3 || lives == 0)
```

RESULT

```
level <= 3 && lives != 0
```

1 Flip the operator: `||` becomes `&&`.

2 Negate each part: `level > 3` becomes `level <= 3`, and `lives == 0` becomes `lives != 0`.

CHECK

```
int level = 2;  
int lives = 3;  
System.out.println(!(level > 3 || lives == 0));  
System.out.println(level <= 3 && lives != 0);
```

OUTPUT

```
true  
true
```

Half a De Morgan is wrong



Negating the parts is only half the job. You must **also** flip `&&` to `||` (or `||` to `&&`).

```
int x = 1;
int y = -1;
System.out.println(!(x > 0 && y > 0));
System.out.println(x <= 0 && y <= 0);
System.out.println(x <= 0 || y <= 0);
```

OUTPUT

```
true
false
true
```

Output line 1 is the original. Line 2 negated the parts but kept `&&`. Line 3 is the correct rewrite.

Push the ! inside

PAUSE AND TRY

Rewrite each condition so it has no ! in front of the parentheses.

	Condition	Your rewrite
1	<code>!(isOpen && hasRoom)</code>	
2	<code>!(age >= 16 hasPermit)</code>	
3	<code>!(count != 0 && total > 100)</code>	

A white rounded square with a teal number '3' inside, tilted slightly to the right.

3

PART 3

Same object, or same text?

KEY IDEA

== on objects asks: the same object?

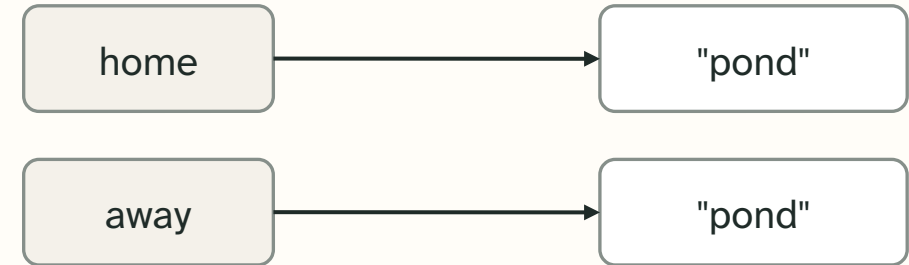
```
String home = new String("pond");  
String away = new String("pond");  
System.out.println(home == away);  
away = home;  
System.out.println(home == away);
```

OUTPUT

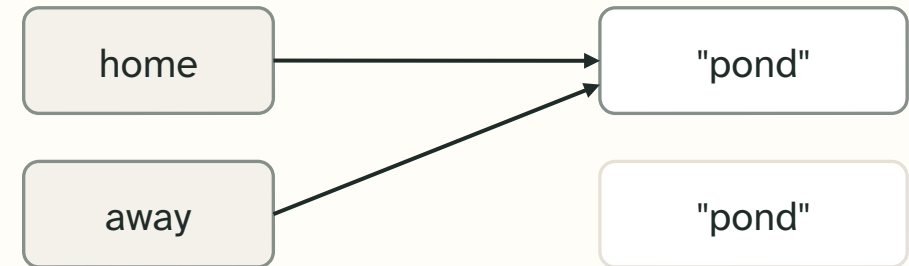
```
false  
true
```

Two variables that refer to the same object are **aliases**.

BEFORE



AFTER away = home;



no longer used

Never compare text with ==



== asks "same object?" To ask "same characters?", call equals.

WITH ==

```
String typed = new String("hop42");
String saved = new String("hop42");
if (typed == saved) {
    System.out.println("Welcome");
} else {
    System.out.println("Try again");
}
```

OUTPUT Try again

WITH EQUALS

```
String typed = new String("hop42");
String saved = new String("hop42");
if (typed.equals(saved)) {
    System.out.println("Welcome");
} else {
    System.out.println("Try again");
}
```

OUTPUT Welcome

Each class decides what "equal" means

Comparison	Question it answers	For two Strings, true when...
<code>a == b</code>	Same object?	a and b hold the same reference
<code>a != b</code>	Different objects?	a and b refer to different objects
<code>a.equals(b)</code>	Same contents?	the characters match exactly



Classes often define their own `equals` method, which decides when two objects count as equivalent, usually by comparing their attributes. `String`'s `equals` compares the characters.

Is anybody there? Check for null



`x == null` is true when `x` refers to **no object**. Check it **before** calling a method, or the program crashes with a `NullPointerException`.



An empty string `" "` is **not** `null`. It is a real object that has no characters.

```
String pick = null;
System.out.println(pick == null);
System.out.println(pick != null
    && pick.equals("blue"));
pick = "";
System.out.println(pick == null);
System.out.println(pick.length());
```

OUTPUT

```
true
false
false
0
```

What prints?

PAUSE AND TRY

Write what each line prints, or "crash" if it crashes.

```
String a = new String("hop");  
String b = a;  
String c = null;  
/* A */ System.out.println(a == b);  
/* B */ System.out.println(a.equals(new String("hop")));  
/* C */ System.out.println(c == null || c.length() > 2);  
/* D */ System.out.println(c.length() > 2 || c == null);
```

Words to know

Equivalent

Two Boolean expressions with the same value in every case.

Truth table

A table of every combination of true and false values.

De Morgan's laws

$\neg(a \ \&\& \ b)$ is $\neg a \ || \ \neg b$, and
 $\neg(a \ || \ b)$ is $\neg a \ \&\& \ \neg b$.

Alias

A second variable that refers to the same object.

null

The value meaning a variable refers to no object.

equals

A method that decides when two objects count as equivalent.

Same, or equal?



Tell your partner one way two conditions can be "the same," and one way two Strings can be "the same."

SENTENCE STARTER

Two conditions are equivalent when _____. Two Strings are == when _____, but equals when _____.

Next: the Same or Equal? puzzle worksheet.