



Computer Science Virtual Learning

HS Computer Science A

April 29th, 2020



Lesson: Short Circuit Evaluation & DeMorgan's Laws

Objective/Learning Target:

Understanding what a short circuit evaluation is and how to
apply it using Java

Understanding DeMorgan's Laws



What is Short Circuit Evaluation?

Both `&&` and `||` use **short circuit evaluation**. That means that the second condition isn't necessarily checked if the result from the first condition is enough to tell if the result is true or false. In a complex conditional with a logical and (`&&`) both conditions must be true, so if the first is false, then the second doesn't have to be evaluated. If the complex conditional uses a logical or (`||`) and the first condition is true, then the second condition won't be executed, since only one of the conditions needs to be true.

In a complex conditional using a logical and (`&&`) the evaluation will short circuit (not execute the second condition) if the first condition is false. In a complex conditional using a logical or (`||`) the evaluation will short circuit if the first condition is true.



Practice

What is printed when the following code executes and x has been set to zero and y is set to 3?

```
if (x == 0 || (y / x) == 3)
    System.out.println("first case");
else System.out.println("second case");
```

- first case
- second case
- You will get an error because you can't divide by zero

What is printed when the following code executes and x has been set to negative 1?

```
String message = "help";
if (x >= 0 && message.substring(x).equals("help"))
    System.out.println("first case");
else System.out.println("second case");
```

- first case
- second case
- You will get an error because you can't use a negative index with substring



What is DeMorgan's Laws?

DeMorgan's laws were developed by Augustus De Morgan in the 1800s. They show how to handle the negation of a complex conditional, which is a conditional statement with more than one condition joined by an and (&&) or or (||), such as `(x < 3) && (y > 2)`.

- not (a and b) is the same as (not a) or (not b). In Java this is written as `!(a && b) == !a || !b`
- not (a or b) is the same as (not a) and (not b). In Java this is written as `!(a || b) == !a && !b`

Applying DeMorgan's Laws to `!(x < 3 && y > 2)` yields `!(x < 3) || !(y > 2)` which means that this complex conditional will be true when `(x >= 3 || y <= 2)`.



What is DeMorgan's Laws?

The negation modifies each conditional as shown below.

- $<$ becomes $>=$
- $>$ becomes $<=$
- $==$ becomes $!=$
- $<=$ becomes $>$
- $>=$ becomes $<$
- $!=$ becomes $==$

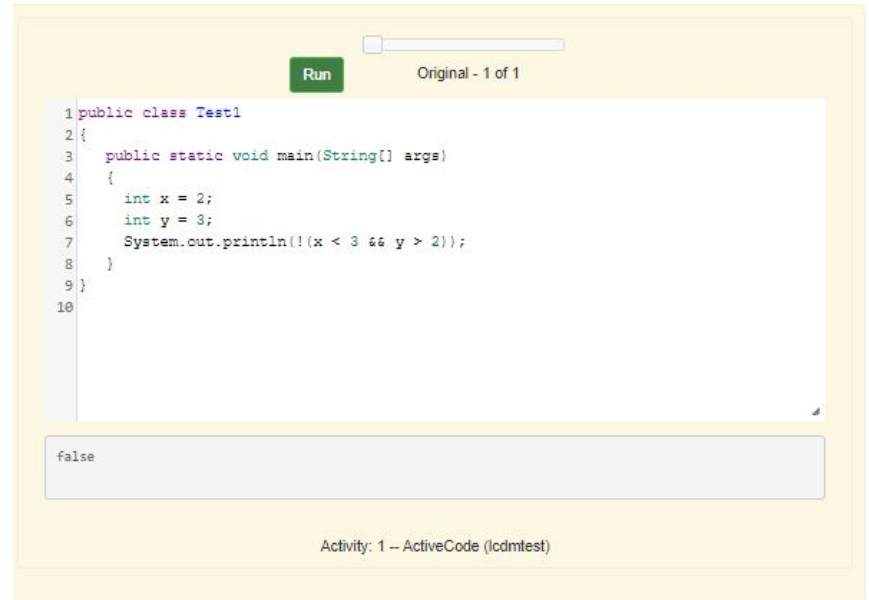
Practice

For what values of x and y will it print true?

Go to:

<https://runestone.academy/runestone/books/published/apcsareview/Conditionals/cDeMorgans.html>

Try out different values of x and y to check your answer.

A screenshot of a Java code editor interface. At the top right, there is a "Run" button and a progress indicator showing "Original - 1 of 1". The code is as follows:

```
1 public class Test1
2 {
3     public static void main(String[] args)
4     {
5         int x = 2;
6         int y = 3;
7         System.out.println(!(x < 3 && y > 2));
8     }
9 }
10
```

Below the code editor, a text box displays the output "false". At the bottom of the editor, it says "Activity: 1 -- ActiveCode (lcmtest)".



Check Your Understanding

1. What is printed when the following code executes and x equals 4 and y equals 3?

```
if (!(x < 3 || y > 2)) System.out.println("first case");  
else System.out.println("second case");
```

- a. first case
- b. second case

2. What is printed when the following code executes and x equals 4 and y equals 3?

```
if (!(x < 3 && y > 2)) System.out.println("first case");  
else System.out.println("second case");
```

- a. first case
- b. second case

3. Which of the following is the same as the code below?

```
!(x > 2 && y < 4)
```

- a. $(x < 2) \parallel (y > 4)$
- b. $(x < 2) \&\& (y > 4)$
- c. $(x \leq 2) \parallel (y \geq 4)$
- d. $(x \leq 2) \&\& (y \geq 4)$

4. Which of the following is the same as the code below?

```
!(x == 2 && y > 4)
```

- a. $(x \neq 2) \parallel (y < 4)$
- b. $(x \neq 2) \&\& (y < 4)$
- c. $(x \neq 2) \&\& (y \leq 4)$
- d. $(x \neq 2) \parallel (y \leq 4)$



For More Resources and to Check Answers

Go to: <https://runestone.academy/runestone/books/published/apcsareview/Conditionals/cShortCircuit.html>

<https://runestone.academy/runestone/books/published/apcsareview/Conditionals/cDeMorgans.html>