

Title: Logical Operators: OR “||”

By Connor Blaser

Assignment Definition and General Feedback By Michael Panitz
at Cascadia Community College (<http://www.cascadia.edu>)

Table of contents:

- [Quick Summary](#) *
- [When To Use \(or Avoid\) This](#) *
- [Example Of Usage](#) *
- [Example Trace](#) *
- [Syntax Explanation](#) *
- [Help With The Logic](#)
- [Important Details](#)
- [Common Mistakes \(And How To Fix Them\)](#)
- [Hyperlinks \(if any\)](#)

Quick Summary:

In Java, you can use the “OR” statement whenever the robot enters a situation where in order to complete an objective, the robot may encounter different situations or need to make a choice on which actions to carry out based on what obstacles are in its way.

When To Use This / Avoid This:

One example of when an OR statement would be beneficial is if you want the robot to carry out one action if it's in different situations. For example, if you want the robot to spin when it has reached a certain coordinate in the city OR when the robot has found a thing. By using the OR statement you can have the robot do an action if either statement above is true. It enables you to make more complex commands that can be used to navigate your robot more smoothly, such as incorporating them in a while or if/else statement.

One example of when not to use this is if you are unsure of what the robot will encounter or need to do, this would result in an OR statement that neither statements is true, such as if there aren't any things or the robot can't reach that particular coordinate within the map.

Example Of Usage:

Let's say you want the robot to spin around when he reaches a specific avenue or can pick a thing up. To do this, you use the OR command within a “while” or if/else statement etc (go to while or if/else statements section if you are unsure what a “while” or if/else statement is) to make the robot look for either statement to be true..

The “while” statement with the OR “||” command would look like this...

Line	Code
1.	<code>import becker.robots.*;</code>
2.	
3.	<code>public class starting_template extends Object</code>
4.	<code>{</code>
5.	<code>public static void main(String[] args)</code>
6.	<code>{ City toronto = new City();</code>
7.	<code>Robot Jo = new Robot(toronto, 3, 0, Direction.EAST, 0);</code>
8.	<code>New Thing(Toronto,3,5);</code>
9.	<code>\\ command code starts here</code>
10.	
11.	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>
12.	<code>{</code>
13.	<code>Jo.move();</code>
14.	<code>}</code>
15.	<code>Jo.turnLeft();</code>
16.	<code>Jo.turnLeft();</code>
17.	<code>Jo.turnLeft();</code>
18.	<code>Jo.turnLeft();</code>
19.	<code>}}</code>

Example Trace:

In a nutshell, this technique allows you to make the robot execute a command when he encounters multiple situations instead of writing a command for every possible situation the robot could encounter.

In order to go over these details more thoroughly, here is a (partial) trace of the above program, with some additional explanation afterwards

Line#	Program Statement	Bob St #	Jo Ave #	Jo Dir	Jo Backpack	True False	Thing
5	<code>public static void main(String[] args)</code>	-	-	-	-	-	-
6	<code>{ City toronto = new City();</code>	-	-	-	-	-	-
7	<code>Robot Jo = new Robot(toronto, 3, 0, Direction.EAST, 0);</code>	3	0	E	0	-	-
8	<code>New Thing (3,5)</code>	3	0	E	0	-	3,5
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	0	E	0	True	3,5
13	<code>Jo.move();</code>	3	1	E	0	-	3,5
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	1	E	0	True	3,5
13	<code>Jo.move();</code>	3	2	E	0	-	3,5
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	2	E	0	True	3,5
13	<code>Jo.move();</code>	3	3	E	0	-	3,5
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	3	E	0	True	3,5
13	<code>Jo.move();</code>	3	4	E	0	-	3,5
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	4	E	0	True	3,5
13	<code>Jo.move();</code>	3	5	E	0	-	3,5
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	5	E	0	False	3,5
15	<code>Jo.turnLeft();</code>	3	5	N	0	-	3,5
16	<code>Jo.turnLeft();</code>	3	5	W	0	-	3,5
17	<code>Jo.turnLeft();</code>	3	5	S	0	-	3,5
18	<code>Jo.turnLeft();</code>	3	5	E	0	-	3,5
Option 2	<code>\\ or if the thing was in a different spot...(starting from the last line 11) this would be the other outcome...</code>	-	-	-	-	Thing changes to 3,8	
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	5	E	0	True	3,8
13	<code>Jo.move();</code>	3	6	E	0	-	3,8
11	<code>while(! Jo.canPickThing() Jo.getAvenue() == 6)</code>	3	6	6	0	False	3,8
15	<code>Jo.turnLeft();</code>	3	6	N	0	-	3,8
16	<code>Jo.turnLeft();</code>	3	6	W	0	-	3,8

17	Jo.turnLeft();	3	6	S	0	3,8
18	Jo.turnLeft();	3	6	E	0	3,8

As shown in the trace, the OR statement is used to make the robot execute an action (spin) when either one of the statements is true. So if the robot reaches avenue 6 it spins OR if it is standing on a thing. The robot will also spin if both statements are true at the same time, for example, if the thing was on avenue 6 then both statements would be true simultaneously and the robot would spin. When using an OR statement, figure out what two possible statements may be true throughout the life of the robot and the OR statement can be used to sync on command to multiple outcomes.

Syntax Explanation:

Note on syntax of command:

Let's start with the program as it's written here.

Line #	Program Source Code
1.	<code>import becker.robots.*;</code>
2.	
3.	<code>public class basic OR extends Object</code>
4.	<code>{</code>
5.	<code>public static void main(String[] args)</code>
6.	<code>{ City toronto = new City();</code>
7.	<code>Robot Jo = new Robot(toronto, 3, 0, Direction.EAST,</code> <code>0);</code>
8.	<code>New Thing (3,5)</code>
9.	<code>\\ command code starts here</code>
10.	
11.	

Below, you can see the finished program with the differences highlighted in yellow, so it's easy to see what's been added/changed). The red highlight is the OR command symbol imbedded in the "while" statement.

Line #	Program Source Code
1.	<code>import becker.robots.*;</code>
2.	
3.	<code>public class basic OR extends Object</code>
4.	<code>{</code>
5.	<code>public static void main(String[] args)</code>
6.	<code>{ City toronto = new City();</code>

7.	Robot Jo = new Robot(toronto, 3, 0, Direction.EAST, 0);
8.	New Thing (3,5)
9.	\\ command code starts here
10.	
11.	while(! Jo.canPickThing() Jo.getAvenue() == 6)
12.	{
13.	Jo.move();
14.	}
15.	Jo.turnLeft();
16.	Jo.turnLeft();
17.	Jo.turnLeft();
18.	Jo.turnLeft();
19.	}}

The OR statement is inserted into a while or if statement inside the parentheses and is used to separate the two (or more) statements. There is no need for semi colons or spacing just use it to divide all the different statements you want the robot to look for.

Help With The Logic:

This is good to use the OR statement whenever your robot needs to do the same thing in different kinds of situations. He will execute the command when either one of the true statements are true as well as when both are true.

Important Details:

- Point #1: when making an OR statement, make sure that at least one of the two statements within the OR statement will be true somewhere in the robots path.
- Point #2: the statement is still considered true if both sides of the OR statement is true simultaneously. In the example above, the statement would still be true if the thing was placed on (3,6).

Mistakes People Commonly Make (And How To Fix Them):

- **Quick Name Of Mistake: permanently false statement**
Detailed Example Of Error: neither side of the OR statement is true.
Detailed Example Of Fix: (unless it's obvious): make sure that at least one side of the statement is true in order to make the OR statement true.

Licensing



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Plagiarism

If you believe that some or all of this document infringes on your intellectual property (i.e., part or all of this document is copied from something you've written) please immediately contact Mike Panitz at Cascadia Community College (perhaps using the Faculty And Staff Directory at <http://www.cascadia.edu/pages/searchtemplate.aspx>)