

Title: Basic If Statement

Author: Farhood Mogharrabin

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](#) *

(Required sections have a * after them in the above list)

Quick Summary:

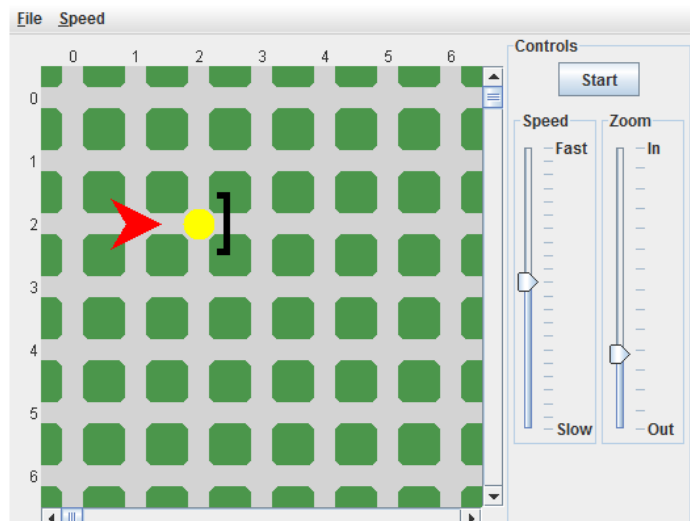
An “If statement” is a feature of the java programming language which is used to check a condition and if that condition is met then it performs a specific operation. If it is not met then nothing happens and the program moves on to the next line of code. Sometimes when you want to perform a conditional statement for different decisions in your program you can use the “if statement”. Basically, when you use the “if statement” in your program, the program asks a question that the answer is true or false and based on the answer makes a decision.

When to Use This:

You can use the “if statement” when the condition is not steady. For instance, in the program of Robot and the City whenever you randomly generate the things and walls you use the “if statement” and give the command to the Robot to move, if front is clear or pick the thing if there is the thing there to pick up.

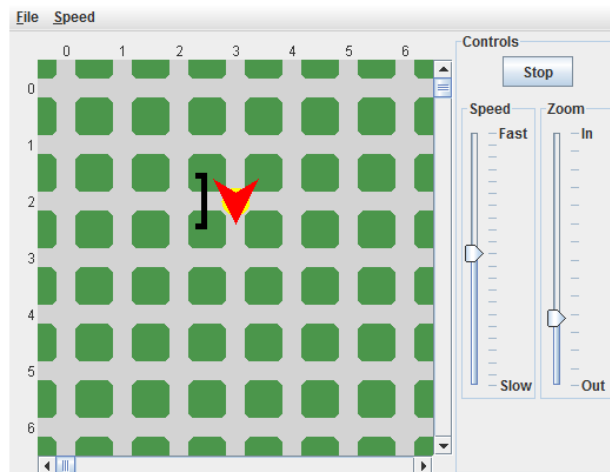
Example of Usage:

Let’s say that we have a city like the picture on the right side of the page. We want to ask the robot to pick the thing and take it behind the wall and put it there.



At first, we give commands to the robot to move one step and if he can pick the thing up, he picks it up.

At the next step, we give another command to the robot that if the front is not clear then it will go around the wall. For this command we made a new service called moveSafe. With this new command the robot checks its way, if it is not clear he will move around the wall by using basic commands such as turn left and move. Picture below shows the robot at the end.



Line	Code
1.	<code>import becker.robots.*;</code>
2.	<code>class RobotMoveSafer extends Robot</code>
3.	<code>{</code>
4.	<code>public RobotMoveSafer(City c, int st, int ave, Direction dir, int num)</code>
5.	<code>{</code>
6.	<code>super(c, st, ave, dir, num);</code>
7.	<code>}</code>
8.	<code>public void moveSafe()</code>
9.	<code>{</code>
10.	<code>if(!this.frontIsClear())</code>
11.	<code>{</code>
12.	<code>this.turnLeft();</code>

13.	<code>this.move();</code>
14.	<code>this.turnLeft();</code>
15.	<code>this.turnLeft();</code>
16.	<code>this.turnLeft();</code>
17.	<code>this.move();</code>
18.	<code>this.turnLeft();</code>
19.	<code>this.turnLeft();</code>
20.	<code>this.turnLeft();</code>
21.	<code>this.move();</code>
22.	<code>}</code>
23.	<code>}</code>
24.	<code>}</code>
25.	<code>public class IfStatement extends Object</code>
26.	<code>{</code>
27.	<code>public static void main(String[] args)</code>
28.	<code>{</code>
29.	<code>City bothell = new City();</code>
30.	<code>RobotMoveSafer ian = new RobotMoveSafer(bothell, 2, 1, Direction.EAST, 0);</code>
31.	<code>new Thing(bothell, 2,2);</code>
32.	<code>new Wall(bothell,2,2, Direction.EAST);</code>
33.	<code>ian.move();</code>
34.	<code>if(ian.canPickThing())</code>
35.	<code>{</code>
36.	<code>ian.pickThing();</code>
37.	<code>}</code>
38.	<code>ian.moveSafe();</code>
39.	<code>ian.putThing();</code>

40.	}
41.	}

Example Trace:

Now that we studied the example of conditional statement, let's have a look at the trace of it. We left the constructor out for clarity.

Line#	Program Statement	Ian St #	Ian Ave #	Ian Dir	Ian Backpack	Thing	Wall	True/false
33	ian.move();	2	2	E	0	2,2	2,2,E	-
34	if(ian.canPickThing())	2	2	E	0	2,2	2,2E	True
36	ian.pickThing();	2	2	E	1	-	2,2E	-
38	ian.moveSafe();	2	2	E	1	-	2,2,E	-
8	public void moveSafe()	2	2	E	1	-	2,2E	-
10	if(!this.frontIsClear())	2	2	E	1	-	2,2E	True
12	this.turnLeft();	2	2	N	1	-	2,2,E	-
13	this.move();	1	2	N	1	-	2,2E	-
14	this.turnLeft();	1	2	W	1	-	2,2E	-
15	this.turnLeft();	1	2	S	1	-	2,2,E	-
16	this.turnLeft();	1	2	E	1	-	2,2E	-
17	this.move();	1	3	E	1	-	2,2,E	-
18	this.turnLeft();	1	3	N	1	-	2,2E	-
19	this.turnLeft();	1	3	W	1	-	2,2E	-
20	this.turnLeft();	1	3	S	1	-	2,2,E	-
21	this.move();	2	3	S	1	-	2,2E	-
39	ian.putThing();	2	3	S	0	2,3	2,2E	-

You'll notice that the trace starts at the beginning of the **main** function on line 38, like normal. On line 39, it is asking a question that if Ian can pick up the thing (“if(ian.canPickThing())”), the answer will return back true. So the program will perform the ian.pickThing command and pick up the thing. For the command on line 43 you'll see that the program automatically returns to line 10 and uses the new service that we made.

Syntax Explanation:

For using the “If Statement” there are a few things that you have to watch out for. The first thing that you have to pay attention to is to write “if” in lower case. Second thing that you have to watch for are parenthesis and curly braces. After the “if”, you have to write your command between parentheses and also you have to make sure that your command has open and close parentheses at the end.

if(ian.canPickThing())

After you write your command you should write the action that you want the robot do in a pair of curly braces. Another important thing that you always have to watch for is to use the name of the robot in your “if statement” unless you are making a new command which in that case you should use “this.”

```
public void moveSafe()
{
    if(!this.frontIsClear())
    {
        this.move();
    }
}
```

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>)