

Title: For Loops

By: Tim Lang

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](#) *

Quick Summary:

In Java, you can use a “for” loop (statement) when there is a certain number times that you want a command or set of commands repeated. A “for” loop is somewhat of an abbreviated while loop in that the syntax is rather close in design but, as opposed to a “while” loop, a “for” loop does not beg to answer a true or false query like a while loop does. A “for” loop simply repeats a set of desired commands a specified and necessary number of times to achieve a desired outcome.

When To Use This / Avoid This:

There are several situations where you will want to use this technique. The only question that the programmer needs to ask is “how many times a command must be executed to get the desired result?” A “for” loop can be used for a simple tasks such as moving forward an absolute number of times, turning an absolute number of times or even picking up and putting things down an absolute number of times depending on what the situation calls for. “For” loops may also be used in non-robot programs as well pertaining to certain situation such as when a program asks for user input and how or whether the program will execute that input data.

There are several situations where you will want to avoid this technique. For instance, in the case of a city that variably changes shape and size, you are never really certain how many times you should move forward or when you need to turn etc. In this case, using a “for” loop may not be a good idea because you don’t know for sure how many times you can or need to move forward or turn without a robot breaking or facing an undesirable direction. In this case, a while loop or an if statement may be more suitable in forming a true or false query based on the question of whether or not the robot can or should move rather than the “for” loop which generally answers a true or false query regarding when a certain number or criteria has been met.

Example Of Usage:

Let's say you need to move 5 times down the side of the city and you know for certain that there is a thing you need to pick up at the 5th intersection. Instead of typing out the move command 5 times and the pickThing command at the end, you could easily and quickly achieve the same goal by using a “for” loop to count out the 5 times that you need to move and then add the pickThing command after the loop end brace.

There are six steps:

First, you need to create the “for” loop. To do this, you need to start by typing out the word “for” which is recognized by the java code as the beginning of a “for” loop.

Second, you need to create the counter inside of the parenthesis that comes after the “for” statement. Example: `for (int ...)` where the ... is where you would place the identifier (initialization) of your counter which can be any word or set of words that describe what it is you are intending to count.

Third, you need to set the counter to whatever integer, Boolean or double that you want the counter to start at.

Fourth, you need to input the number of times that you wish the program to execute the intended action, also known as the “limit” or “check”. This can be done by using the counter identifier and then a greater than, less than, equal to or not equal to symbol followed by the number of times you wish the command to be executed.

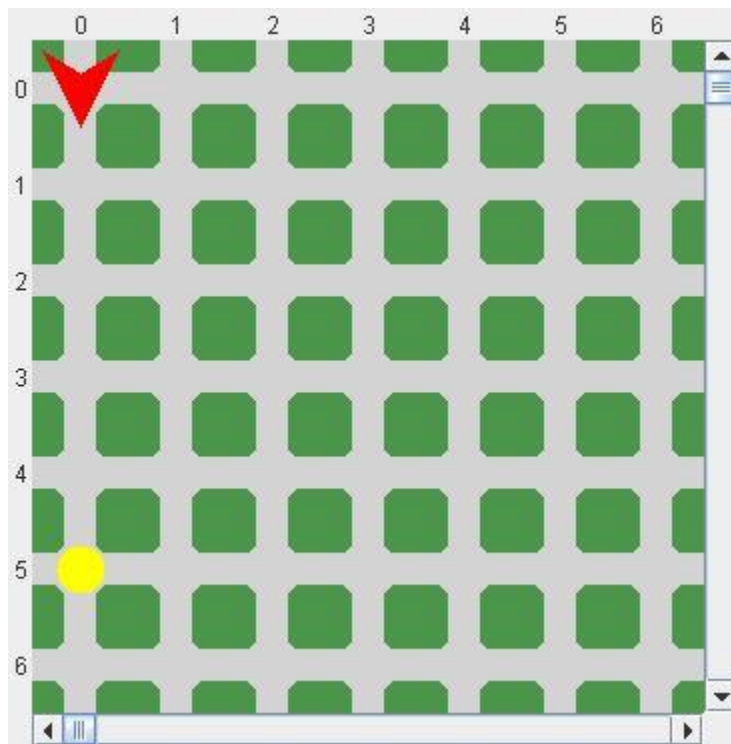
Fifth, you add the part of the “for” loop statement which states how many times the loop has been completed also known as the “update”. To do this you use the counter (identifier) to equal the counter plus whatever it is that you want the count to be at the end of the loop.

Sixth, you add the “statement to repeat” inside the loop brackets (braces) which are the commands to be executed each time the loop occurs.

Line	Code
1.	<code>import becker.robots.*;</code>
2.	
3.	<code>public class Example_for_loops extends Object</code>
4.	<code>{</code>
5.	<code>public static void main(String[] args)</code>
6.	<code>{</code>
7.	<code>City loopland = new City();</code>
8.	<code>Robot bob = new Robot (loopland, 0, 0, Direction.SOUTH, 0);</code>
9.	<code>new Thing (loopland, 5, 0);</code>

10.	
11.	<code>for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)</code>
12.	<code>{</code>
13.	<code>bob.move();</code>
14.	<code>}</code>
15.	<code>bob.pickThing();</code>
16.	<code>}</code>
17.	<code>}</code>

Below is an example of the program that the above syntax has initially created, before the “for” loop has begun:



Next, I will provide an example trace of the program and provide a final “picture” of what the program would look like after the “for” loop has completed and the program comes to an end.

Example Trace:

In a nutshell, this technique allows you to create a sequence or loop that will count up to a certain number that the user provides in order to obtain a desired amount of times a command or set of commands are executed.

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 #	Bob Ave #	bob Dir	Bob Backpack Contents	Thing Location	True or False	Counter
5	<code>public static void main(String[] args)</code>	-	-	-	-	-	-	-
7	<code>City loopland = new City();</code>	-	-	-	-	-	-	-
8	<code>Robot bob = new Robot (loopland, 0, 0, Direction.SOUTH);</code>	0	0	SOUTH	0	-	-	-
9	<code>new Thing (loopland, 5, 0);</code>	0	0	SOUTH	0	(5, 0)	-	-
11	<code>for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)</code>	0	0	SOUTH	0	(5, 0)	True	0
13	<code>bob.move();</code>	1	0	SOUTH	0	(5, 0)	-	0
11	<code>for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)</code>	1	0	SOUTH	0	(5, 0)	-	1
11	<code>for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)</code>	1	0	SOUTH	0	(5, 0)	True	1
13	<code>bob.move();</code>	2	0	SOUTH	0	(5, 0)	-	1
11	<code>for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)</code>	2	0	SOUTH	0	(5, 0)	-	2
11	<code>for (int numToMove = 0; numToMove < 5; numToMove = numToMove</code>	2	0	SOUTH	0	(5, 0)	True	2

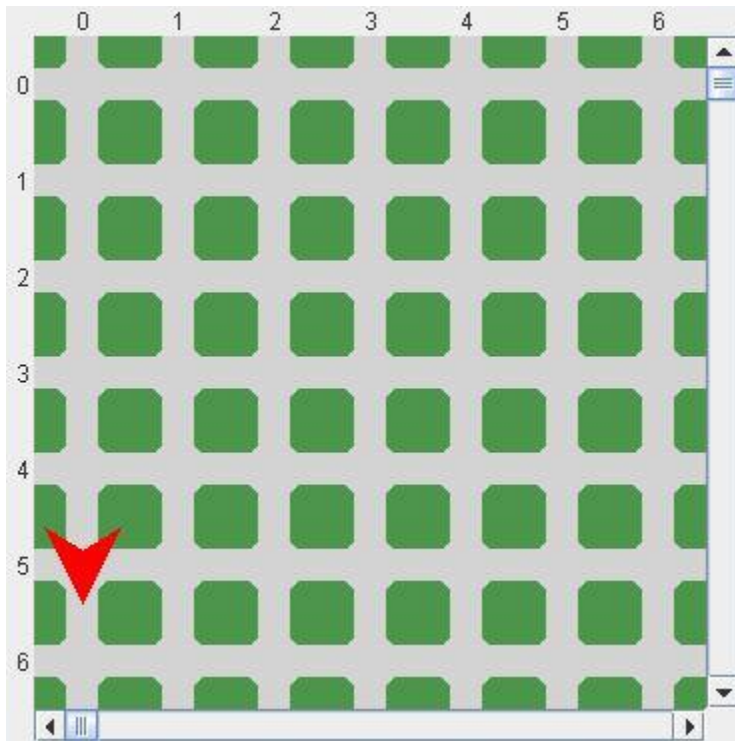
	+1)							
13	bob.move();	3	0	SOUTH	0	(5, 0)	-	2
11	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)	3	0	SOUTH	0	(5, 0)	-	3
11	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)	3	0	SOUTH	0	(5, 0)	True	3
13	bob.move();	4	0	SOUTH	0	(5, 0)	-	3
11	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)	4	0	SOUTH	0	(5, 0)	-	4
11	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)	4	0	SOUTH	0	(5, 0)	True	4
13	bob.move();	5	0	SOUTH	0	(5, 0)	-	4
11	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)	5	0	SOUTH	0	(5, 0)	-	5
11	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)	5	0	SOUTH	0	(5, 0)	False	5
15	bob.pickThing();	5	0	SOUTH	1	-	-	-

You'll notice that the trace starts at beginning of the **main** function, on line 5, like normal. It proceeds normally until line 11 where the “for” loop begins. At this point, lines 11 and 13 are repeated five times because the counter began at 0 and the loop then kept repeating itself until the counter reached 5 and was no longer less than 5. Since the counter reached the integer of 5, the “for” loop had then run its course and the program moved on to line 15.

You will also notice, that in the example trace given above, that line 11 has been repeated in conjunction with which portion of the “for” loop is being processed (executed). What this means

is that in order to understand exactly “how” the loop is functioning and which portion of the “for” loop is being considered at what time, the portions in the repeated line 11 syntax that are highlighted are the portions of the code that are being checked and updated. Also the subsequent True/False answers and counter numbers are highlighted to correspond with the portion of the “for” loop that has or is changing as the program cycles.

Below is an example “picture” of what the program has accomplished after running the “for” loop and subsequent “pickThing” command at the end of the trace:



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 Example for loops extends Object</code>
4.	<code>{</code>
5.	<code>public static void main(String[] args)</code>
6.	<code>{</code>
7.	<code>City loopland = new City();</code>

8.	Robot bob = new Robot (loopland, 0, 0, Direction.SOUTH, 0);
9.	new Thing (loopland, 5, 0);
10.	
11.	bob.move();
12.	bob.move();
13.	bob.move();
14.	bob.move();
15.	bob.move();
16.	bob.pickThing();
17.	}
18.	}

Below, you can see the finished program with the differences highlighted in yellow, so it's easy to see what's been added/changed).

Line #	Program Source Code
1.	import becker.robots.*;
2.	
3.	Public class Example for loops extends Object
4.	{
5.	public static void main(String[] args)
6.	{
7.	City loopland = new City();
8.	Robot bob = new Robot (loopland, 0, 0, Direction.SOUTH, 0);
9.	new Thing (loopland, 5, 0);
10.	{
11.	for (int numToMove = 0; numToMove < 5; numToMove = numToMove +1)
12.	{
13.	bob.move();
14.	}
15.	bob.pickThing();
16.	}
17.	}

The syntax of the basic “for” loop is quite simple and very clean in nature. As you can see in the above scenario, the words numToMove (short for number to move) were used to identify the counter but it’s just as easy and legal to simply use the word “counter”. Example: for (int counter...)

You may set the counter to start at any number you want depending on the circumstances but for the

purpose of executing an action or set of actions a certain number of times, it is best to start at zero because it is easier for you to understand where you are beginning and how many times the loop will commence and run its course. Example: `for (int counter = 0; ...)`

Also, in a “for” loop, you will notice that each portion of the statement ends with semi colons (highlighted in green). The “int” (initial number) must have a semi colon after it, example: `int numToMove = 0;`

as does the check section,

example: `numToMove < 5;`

Essentially, this means that very basic structure of the “for” loop would be `for (; ;)`. Where the best practice is to place the initial number before the first semi colon, the check section before the second semi colon and the update after and between the second semi colon and closing parenthesis bracket respectively.

In the example syntax above, the robot will proceed to move a total of 5 times which is as many times as the loop is set to occur until the counter is no longer less than 5 (i.e. equal to 5).

Example: `for (int numToMove = 0; numToMove < 5;`

In this example, the counter will begin at 0 and continue to loop the statement to repeat as long as the count is less than 5. Every time the loop is completed, the integer amount has been increased by a value of 1; hence, the loop will occur four more times from this point. It is also worth noting, that instead of using +1 at the end of the statement, you may also use ++ which means the same thing as +1. Example: `for (int numToMove = 0; numToMove < 5; numToMove = numToMove + 1)` or `for (int numToMove = 0; numToMove < 5; numToMove++)`

Finally, the statement to repeat which is the command or commands that you would like to be repeated in the “for” loop go inside the braces of the loop and in this case have been looped 5 times thereby moving the robot 5 times.

Example:

```
for (int numToMove = 0; numToMove < 5; numToMove = numToMove + 1)
{
    bob.move();
}
```

Basic syntax of the “for” loop statement:

```
for (int 'counter' = 0; 'counter' < 'limit'; 'counter' = 'counter' + 1)
```

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 this entire document infringes on your intellectual property (i.e., part or this entire 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>)