

Getting User Input Using The Scanner Object

Brandon Streater

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 the scanner object to receive user input. The first way we learned to use this “input” is with the keyboard. It is important to remember that when running the program to click “start” first. In this example, the program will pause to receive the user input. When you enter the input for the scanner it will process it based on what you tell the program to ask for (character, integer, etc...).

When To Use This / Avoid This:

Using this command is very helpful when asking the user for information to continue or determine where the program goes. For example, creating an online test for workplaces or online schools so that it will tell them if the answer is correct or not and to continue to the next quest. This would also be a very elementary version of how a game would work (“A” to punch and “B” to kick.)

The only situation you would not want to use this would be if there is no reason to ask for user input. If you want the program to run, and that is it, then there is no need for user input.

Example Of Usage:

Line	Code
1.	<code>import becker.robots.*;</code>
2.	<code>import java.util.*;</code>
3.	
4.	<code>public class Write_A_Book extends Object</code>
5.	<code>{</code>
6.	<code>public static void main(String[] args)</code>

7.	{
8.	System.out.println("...Program begins here...");
9.	int numMoves = 0;
10.	int counter = 0;
11.	Scanner keyboard = new Scanner(System.in);
12.	
13.	
14.	City edmonds = new City();
15.	Robot ava = new Robot(edmonds, 1, 1, Direction.SOUTH, 0);
16.	
17.	System.out.println ("How many intersections forward would you like Ava to move?");
18.	if(keyboard.hasNextInt())
19.	{
20.	numMoves = keyboard.nextInt();
21.	System.out.println ("You asked to move " + numMoves + " spaces");
22.	counter = 0;
23.	while(counter < numMoves)
24.	{
25.	ava.move();
26.	counter = counter + 1;
27.	}
28.	
29.	else
30.	{
31.	System.out.println ("Please enter a positive integer.");
32.	}
33.	
34.	keyboard.nextLine();
35.	
36.	System.out.println ("...Program ends here...");
37.	}
38.	}

Example Trace:

In the example trace, we will see what happens when we type into the program a number, and have it run the way it is intended to.

Line#	Program Statement	Robo t's Stree t #	Robot' s Ave #	Robot' s Directi on	Items In Backp ack	Input (keyb oard)	Output	num Move s	counter	True/ False
8	<code>System.out.println("... Program begins here...");</code>	-	-	-	-	-	...Program begins here...	-	-	-
10	<code>int numMoves = 0;</code>	-	-	-	-	-	-	0	-	-
11	<code>int counter = 0;</code>	-	-	-	-	-	-	0	0	-
12	<code>Scanner keyboard = new Scanner(System.in);</code>	-	-	-	-	-	-	0	0	-
14	<code>City edmonds = new City();</code>	-	-	-	-	-	-	0	0	-
15	<code>Robot ava = new Robot(edmonds, 1, 1, Direction.SOUTH, 0);</code>	1	1	S	0	-	-	0	0	-
17	<code>System.out.println ("How many intersections forward would you like Ava to move?");</code>	1	1	S	0	-	How many intersecti ons forward would you like Ava to move?	0	0	-
18	<code>if(keyboard.hasNextInt())</code>	1	1	S	0	1 <ent er>	-	0	0	True
20	<code>numMoves = keyboard.nextInt();</code>	1	1	S	0	-	-	1	0	-

21	<code>System.out.println ("You asked to move " + numMoves + " spaces");</code>	1	1	S	0	-	You asked to move 2 spaces	1	0	-
22	<code>counter = 0;</code>	1	1	S	0	-	-	1	0	-
23	<code>while(counter < numMoves)</code>	1	1	S	0	-	-	1	0	True
25	<code>ava.move();</code>	2	1	S	0	-	-	1	0	-
26	<code>counter = counter + 1;</code>	2	1	S	0	-	-	1	1	-
23	<code>while(counter < numMoves)</code>	2	1	S	0	-	-	1	1	False
34	<code>keyboard.nextLine();</code>	2	1	S	0	-	-	-	-	-
36	<code>System.out.println ("...Program ends here...");</code>	2	1	S	0	-	...Program ends here...	-	-	-

Secondly I will show a trace that show what happens if we enter an incorrect input such as “B.” Since “B” is not an integer it will not run the same way as if you were to enter a positive integer such as “1.

Line#	Program Statement	Robot's Street #	Robot's Ave #	Robot's Direction	# Items In Backp	Input (keyboard)	Output	num Moves	counter	True/False
-------	-------------------	------------------	---------------	-------------------	------------------	------------------	--------	-----------	---------	------------

ack

8	System.out.println("... Program begins here...");	-	-	-	-	-	...Program begins here...	-	-	-
10	int numMoves = 0;	-	-	-	-	-	-	0	-	-
11	int counter = 0;	-	-	-	-	-	-	0	0	-
12	Scanner keyboard = new Scanner(System.in);	-	-	-	-	-	-	0	0	-
14	City edmonds = new City();	-	-	-	-	-	-	0	0	-
15	Robot ava = new Robot(edmonds, 1, 1, Direction.SOUTH, 0);	1	1	S	0	-	-	0	0	-
17	System.out.println ("How many intersections forward would you like Ava to move?");	1	1	S	0	-	How many intersections forward would you like Ava to move?	0	0	-
18	if(keyboard.hasNextInt())	1	1	S	0	B <enter>	-	0	0	
29	else	1	1	S	0	-	-	0	0	
31	System.out.println ("Please enter a positive integer.");	1	1	S	0	-	Please enter a positive integer.	0	0	
34	keyboard.nextLine() ;	1	1	S	0	-	-	0	0	

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.	<code>import java.util.*;</code>
3.	
4.	<code>public class Write A Book extends Object</code>
5.	<code>{</code>
6.	<code>public static void main(String[] args)</code>
7.	<code>{</code>
8.	<code>System.out.println("...Program begins here...");</code>
9.	<code>int numMoves = 0;</code>
10.	<code>int counter = 0;</code>
11.	<code>Scanner keyboard = new Scanner(System.in);</code>
12.	
13.	
14.	<code>City edmonds = new City();</code>
15.	<code>Robot ava = new Robot(edmonds, 1, 1, Direction.SOUTH, 0);</code>
16.	
17.	<code>System.out.println ("How many intersections forward would you like Ava to move?");</code>
18.	<code>if(keyboard.hasNextInt())</code>
19.	<code>{</code>
20.	<code>numMoves = keyboard.nextInt();</code>
21.	<code>System.out.println ("You asked to move " + numMoves + " spaces");</code>
22.	<code>counter = 0;</code>
23.	<code>while(counter < numMoves)</code>
24.	<code>{</code>
25.	<code>ava.move();</code>
26.	<code>counter++;</code>
27.	<code>}</code>
28.	
29.	<code>else</code>
30.	<code>{</code>
31.	<code>System.out.println ("Please enter a</code>

	<code>positive integer.");</code>
32.	<code> }</code>
33.	
34.	<code> keyboard.nextLine();</code>
35.	
36.	<code> System.out.println ("...Program ends here...");</code>
37.	<code> }</code>
38.	<code>}</code>

import java.util.*;

It is important to add this to the beginning of any program using the scanner function. This allows the program to make use of commands in java.util. Much like becker.robots for the "turnLeft" command.

Scanner keyboard = new Scanner(System.in);

Here we are giving a name to the scanner variable. In this case, we are giving it the name "keyboard."

System.out.println ("How many intersections forward would you like Ava to move?");

Adding this line tells the user what kind of command it is looking for. Otherwise your program would pause looking for what, a number? a letter? true/false?

if(keyboard.hasNextInt())

This tells the program to wait for your command.

Lines 20-32

This is the while loop that tells the program what to do with your keyboard input. It assigns the name numMoves to the input and tells the robot (ava) to move while the "counter" is less than your input or "numMoves." If a positive integer is not provided, it falls to the "else" portion of the loop.

keyboard.nextLine();

This is crucial to add as it clears whatever input was provided before. Without this, there is a possibility of an infinite loop or if you ever add another scanner input to your program.



Getting User Input Using The Scanner Object by Brandon Streater is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/).

Note that you must choose ONE option in this section (share or not), but which option you choose will have no affect on your grade.

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>)