Title: Using System.out.print, and println

Author # Rebecca Lebbos

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

Table of contents:

Quick Summary:

In Java *System.out.print* and *System.out.println* are used to print whatever is inside the parentheses in the compile messages while the program is running. The information printed can be variables or expressions that help readers understand what the program is doing at a certain time. It provides important details about when the program is starting, ending, and more details about where the robot is at and what it's doing. In other words the System.out.print is a communicator.

When To Use This / Avoid This:

The System.out.print( ) is used in a Java script to print the desired information while running the program.

There are two methods System.out.print( ) prints anything in the parentheses on the line as the last system.out.print. The second method is system.out.println( ) it print everything in the parentheses on the next line, the "ln" represents line.

There are no situations where you will want to avoid this technique unless it is unnecessary.

Example Of Usage:

Let's say we have a robot called me and he wants to move around the city get the things and go back to his house. We want to keep track of his every move, the readers want to know where the program starts, what the robot is doing and his coordinates every time he moves and turns.

There are three steps for this process
First, we have to add import Java.util.*; on the top on the page line 2, then we need to create the city, robot, things and walls. Note that this is the first step in creating any kind of java program with Robots and cities.

Second we want to use the system.out.print after each command. To make it easier for us we can create a method like the one on line 8 “displayCoordinates( )” to plug in after each command to get the coordinates. On line 34 the system.out.println is used to mark where the robot Me’s job begins. On lines 44,54 and 64 the system.out.println( ) is used to clarify the turns me is making to explain the three similar coordinates as he turns left three times.

Third we want to create the code to make me finish his job and return home.

When we run the program a message like the one below will appear explaining the system:

```
Ï«Ï ----jGRASP exec: Java simple_example
ÏÏ§Ï
ÏÏ§Ï me's job begins here   ( line 34)
ÏÏ§Ïstreet:4 ave:1
ÏÏ§Ïstreet:5 ave:1
ÏÏ§Ï me is turning left, Direction, East
ÏÏ§Ïstreet:5 ave:1
ÏÏ§Ïstreet:5 ave:2
ÏÏ§Ïstreet:5 ave:3
ÏÏ§Ï me is turning left, Direction, North
```

```
  1     import becker.robots.*;
  2     import Java.util.*; // step one
  3      public class simple_example extends Object
  4     {
  5     //step 2
  6   // so we want to know the location of the Robot everytime it moves or turns.
  7     // so we create a method called displayCoordinates so we don't have to call it each time it moves.
  8         public static void displayCoordinates(Robot me)
  9        {
 10           int ave;
 11           int street;
 12
```

```

          ave = me.getAvenue();
          street=me.getStreet();


System.out.println("street:" + street+ " "  +  "ave:" + ave);
// or can be writin: ( steert+ "
"+ ave)
       }_
    _
        public static void main(String[] args)
       {
          City Sea= new City();
        Robot me = new Robot(Sea,3,1, Direction.SOUTH,0);

        new Thing(Sea, 3, 1);


        new Thing(Sea, 2, 2);
        new Thing(Sea, 3, 2);

        new Wall(Sea,2,2,Direction.EAST);
        new Wall(Sea,1,2,Direction.SOUTH);
        new Wall(Sea, 3, 2, Direction.NORTH);

       //creating the code
 step 3
          System.out.println(" me's job begins here");

          me.move();
          displayCoordinates(me);
       _
          me.move();
          displayCoordinates(me);
          //System.out.println(" me is about to turn left");
       _
          me.turnLeft();
 System.out.println(" me is turning left, Direction East”);
          displayCoordinates(me);
       _
          me.move();
          displayCoordinates(me);
       _
          me.move();
          displayCoordinates(me);
       _
          me.turnLeft();
System.out.println(" me is turning left, Direction North”)
          displayCoordinates(me);

          me.move();
          displayCoordinates(me);

          me.move();
          displayCoordinates(me);

          me.turnLeft();
 System.out.println(" me is turning left, Direction West”);
          displayCoordinates(me);

          me.move();
          displayCoordinates(me);

          me.pickThing();
          displayCoordinates(me);

          me.move();
          displayCoordinates(me);

          me.turnLeft();
          displayCoordinates(me);
System.out.println(" me is turning left, Direction South”);

          me.turnLeft();
          displayCoordinates(me);
 System.out.println(" me is turning left, Direction East”);

          me.turnLeft();
          displayCoordinates(me);
  System.out.println(" me is turning left Direction North”);

          me.pickThing();
          displayCoordinates(me);

          me.move();
          displayCoordinates(me);

          me.turnLeft();
          displayCoordinates(me);
System.out.println(" me is turning left, Direction South”);
```

```
 97
 98            me.turnLeft();
 99            displayCoordinates(me);
100 System.out.println(" me is turning left, Direction East”);
101
102            me.turnLeft();
103            displayCoordinates(me);
104System.out.println(" me is turning left, Direction North”);
106            me.move();
107            displayCoordinates(me);_
108        _
109            me.pickThing();_
110        _
111            System.out.println( "me's job is complete");
112
113        }
114    }
```

Note: The turnRight() command was not used to make things more interesting. In a nutshell, this technique allows you to track the program and understand it better.

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	me St #	me Ave #	Me Dir	Bob Backpack	Thing 1loc	Thing 2 loc	Thing 3 location	System Output
34	System.out.println(" me's job begins here");	4	1	S	_	(3,1)	(2,2)	(3,2)	Me’s job begins here
36	me.move();	4	1	S	-	(3,1)	(2,2)	(3,2)	-
37	displayCoordinates(me);	4	1	S	-	(3,1)	(2,2)	(3,2)	street:4 ave:1
8	public static void displayCoordinates(Robot me)								
10	int ave;	-	-	-	-	-	-	-	–
11	int street;	-	-	-	-	-	-	-	–
14	ave = me.getAvenue();	-	4	-	-	-	-	-	–
15	street=me.getStreet();	1	-	-	-	-	-	-	–
18	System.out.println("str								street:4 ave:1

	`eet:" + street+ " "  + "ave:" + ave);`								
39	`me.move();`	5	1	S	-	(3,1)	(2,2)	(3,2)	-
40	`displayCoordinates(me);`	5	1	S	-	(3,1)	(2,2)	(3,2)	street:5 ave:1
8	`public static void displayCoordinates(Robot me)`								
10	`int ave;`	-	-	-	-	-	-	-	–
11	`int street;`	-	-	-	-	-	-	-	–
14	`ave = me.getAvenue();`	-	1	-	-	-	-	-	–
15	`street=me.getStreet();`	5	-	-	-	-	-	-	–
18	`System.out.println("street:" + street+ " "  + "ave:" + ave);`		-	-	-	(3,1)	(2,2)	(3,2)	street:4 ave:1
43	`me.turnLeft();`	5	1	E	-	(3,1)	(2,2)	(3,2)	-
44	`System.out.println(" me is turning left, Direction, East ");`	5	1	E	-	(3,1)	(2,2)	(3,2)	Me is turning left, Direction, East
45	`displayCoordinates(me);`	5	1	E	-	(3,1)	(2,2)	(3,2)	street:5 ave:1

You'll notice that the trace begins when me's job bgins, and the wall are not included.

Note as well that when the desired text and numbers print out the System.out.println does not print the closing and opening " " that's because they are used only in the Java code writing.

Syntax Explanation:

Note on syntax of command: The system.out.print () and .println() print anythings that is inside the parentheses. The sentences that we want to print as they are we have to include them in between the " " so let's say we want to print "Hi everyone", we do system.out.println (" Hi everyone"), the system recognizes everything inside the parentheses as long as it's place in between the " " inkling space and dots or anything we please to print. In order to print multiple sentences or sentences followed by number they need to be separated by a + sign inside the parentheses like: System.out.println("street:" + street+ " " + "ave:" + ave);. Not that the words that are not included in between the “ “ are printed as integers that were previously defined in the beginning of the program. There are different correct ways to write this command for example use ++ like , `( steert+ ","+ ave).`

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

```
    import becker.robots.*;
     import Java.util.*;
 _
 _ public class explenation extends Object
     {
         public static void main(String[] args)
        {
           System.out.println("THE PROGRAM STARTS HERE!!");
 ___
 ___
 ___City Bey = new City();
 ___Robot hi = new Robot (Bey,3,4,Direction. EAST,0);
 _  new Thing ( Bey, 1, 3);
 ___new Thing ( Bey, 2,3);
 ___Robot you = new Robot(Bey,3,5,Direction.EAST,0);
 ___
 ___you.move();
 ___System.out.println(" you is entering hi's world");
 ___
 ___while ( hi.getAvenue()<= you.getAvenue() )
 ___  {
 ___   you.move();
 ____System.out.println( "going around Hi's world");

 ____you.turnLeft();
 ____System.out.println( "going around Hi's world");
 ____

 ____you.move();
 ____System.out.println( "going around Hi's world");

 ____you.move();
 ____System.out.println( "going around Hi's world");
  }
 ___
 ___System.out.println( "Ooops! he saw me, bye bye !");
 __}
 _}
```

```
___
___
```

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

```
  import becker.robots.*;
    import Java.util.*;
_
_ public class explenation extends Object
    {
         public static void main(String[] args)
        {
           System.out.println("THE PROGRAM STARTS HERE!!");
___
___
___City Bey = new City();
___Robot hi = new Robot (Bey,3,4,Direction. EAST,0);
_       new Thing ( Bey, 1, 3);
___new Thing ( Bey, 2,3);
___Robot you = new Robot(Bey,3,5,Direction.EAST,0);
___
___you.move();
___System.out.println(" you is entering hi's world");
___
___while ( hi.getAvenue()<= you.getAvenue() )
___  {
___   you.move();


____you.turnLeft();
____

____you.move();
_
____you.move();
____System.out.println( "going around Hi's world");
 }
___
___System.out.println( "Ooops! he saw me, bye bye !");
__}
_}

```

The changes made are very simple, instead of having the system print “going around Hi’s world every time you makes move, we can place the System.out.println("going around Hi's world"); at the end of the cycle that way we know that when the command prints the line going around Hi's world one cycle is complete. The system.out.println( ) is used for anything, it may be more complicated and used to count the cycles, or include the I and the numbers each time.

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)