**Java Lesson: Looping  
Last Updated: 2/6/2018  
mr Hanley**

**Objective:** The objective of this lesson is to introduce the student to the java commands for looping to accomplish a particular algorithm.

We are often finding ourselves defining algorithms where a certain set of steps is repeated over and over again. This is known in computer science as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Various commands are available

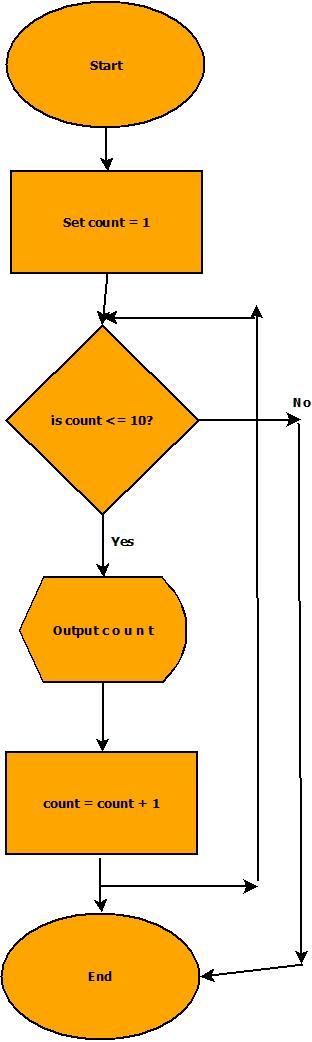
the for loop, the do..while loop and the while loop are typical in programming languages.

Perhaps we want to read in free throw results from a user and compute the ft %. Or maybe we want to print out the headings for a calendar.

## **Contents**

1. Looping a fixed number of times
2. Looping an unknown number of times
3. Using the break command
4. Using counters and accumulators
5. Nested Loops
6. The precious for loop
7. **Fixed number of iterations**

Suppose we want to repeat a command a definite number of times  
Example 1: Printing 10 integers to the console starting at 1



**Practice 1: You try:** Write commands for java to output 100 numbers starting with 1 and increasing by 1.

**Practice 2: You try:** Print your name out to the screen 50 times

**Practice 3: You try:** Print the numbers from 30 down to -2 to the screen on the same line with a space in between

1. **Sometimes we don’t know how many times we want to iterate**

For example, reading in a data file of unknown size.

|  |  |
| --- | --- |
|  | **Must have a file in the project folder called whatever**  **The variable filename is (example rates.txt)**  Euro  .95  Chinese Yuan Renminbi  6.91  Canadian Dollar  1.35  **Java:**  try {  //Attempt to open the file  File f = new File(fileName);  //Assuming its open, let's grab the currencies  Scanner input = new Scanner(f);  while (input.hasNext()) {  //Read the currency  String descript;  descript = input.nextLine();  String temp = input.nextLine();  double convertRate = Double.parseDouble(temp);  //Now create a new Currency object  Currency c = new Currency(descript, convertRate);  //Add into the array  if (numRates < 100) {  rates[numRates] = c;  numRates++;  }  }  input.close();  }  catch (Exception e) {  System.out.println(" ----------------");  System.out.println("| FILE problem |");  System.out.println(" ----------------");  System.out.println(e); //print the excepti  } |

**Practice 4: You try:** Ask the user for a number. Print all the even numbers from 0 up til and including that number

Scanner input = new Scanner(System.in); //Create a Scanner to read user input

You can also cause the loop to count up by a different number or count down if you like

int i = 0;

while(i<20) {

System.out.println(“ i = “ + i);

i = i + 5;

}

will print

i = 0

i = 5

i = 10

i = 15

**Practice 5: you try**:

Print the numbers 1,8,15,22,29 with an empty space in between them using a while loop

1. **Using the break command**

The break command \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Using the break command to exit a menu

while (true) {

System.out.println("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$");

System.out.println(" TO U S $ F R O M ");

System.out.println();

System.out.println(“\u001B[31m”); //This is red, see the cookbook

System.out.println(" 1 = Euros");

System.out.println(“\u001B[34m”); //This is blue, see the cookbook

System.out.println(" 2 = GB Pounds");

System.out.println(“\u001B[36m”); //This is cyan, see the cookbook

System.out.println(" 3 = Japanese Yen");

System.out.println(“\u001B[33m”); //This is yellow, see the cookbook

System.out.println(" 4 = Chinese Yuan Renminbi");

System.out.println(“\u001B[35m”); //This is purple, see the cookbook

System.out.println(" 5 = Mexican Pesos");

System.out.println(“\u001B[32m"”); //This is green, see the cookbook

System.out.println(“\u001B[0m"); //Resets console back to orig color

System.out.println();

System.out.println(" 0 = BACK TO MAIN");

System.out.println("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$");

System.out.println("Enter Choice ->");

char ch = input.next();

if (ch.charAt(0).==(‘0') {

break;

}

} //end while

1. **Using counters and accumulators**

An accumulator:

A counter:

#### Practice 6: Redo project AlgoPract1\_2, the free throw problem. Alter the problem so that the user can type in any number of free throw results and get the % made. Use –1 as a sentinel.

Scanner input = new Scanner(System.in); //Create a Scanner to read user input

1. **Nested Loops**



By nesting one loop inside another, you can accomplish much

int i = 0;

while (i <= 4){

System.out.println(“ i = “ + i);

int j = 0;

while (j < 3){

System.out.println(“ j = “ + j);

j++;

}

i++;

}

will print(fill in the blanks)

i = 0

j = 0

j = 1

j = 2

i = 1

j = 0

j =

j =

i = 2

j = 0

j =

j = 2

i =

j = 0

j = 1

j = 2

i =

j =

j =

j =

**Practice 7: you try** Use nested loops to draw the following image

\*\*\*\*

\*\*\*\*

\*\*\*\*

**Practice 8: you try** Use nested loops to make the following image

^^^^

**\*\*\*\***

**^^^^**

**\*\*\*\***

**^^^^**

**\*\*\*\***

**Practice 9: you try** Print a scalable box as follows;

1

&

2

&&

&&

3

&&&

& &

&&&

4

&&&&

& &

& &

&&&&

**Practice 10: Displaying a maze the user can see in 2 dimensions**

If we want to display a 2d maze on the console, we can use nested loops

Example

int [] [] maze = new int [10][10]; //creates a 2d array of 100 integers, 0 is default value

final int WALL = 1; //made up a code that represents a wall

final int DOOR = 2; //door you can pass through

//Place some walls and doors to make this picture

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

//Do the commands here to place walls

How can we display the maze contents to the screen?

1. **The precious for loop**

The for loop is one of the oldest commands in computer programming

for (*initializer statements; control expression; step expression)*

*{*

*body of loop*

*}*

Since everything is on one line, this is simply a delicious command!

To print the first 10 positive integers(beginning with 1) to the screen, use

for(int i=1; i<11; i++)

{

System.out.println(i);

}

You can also leave any of the parts blank or put multiple statements in the first and third separated by ,

**Practice 11: you try** Write a for loop to print out the numbers from 1 to 1000, every 10 numbers print out a \*

**Practice 12: you try** Write a for loop that will print first 50 powers of 2, ie

1

2

4

8

16

32

…