Building Java Programs

Chapter 2

Lecture 2-3: Nested Loops

reading: 2.3

Write a Java method using a *for* loop to display the following:

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Easy, each line is identical,
Just need to print the 5 asterisks 5 times
…each asterisk is followed by a space…

```
public static void printBlock() {
   // print 5 rows of 5 asterisks (space separated)
   for ( int i = 1 ; i <= 5 ; i++ ) {
      System.out.println("* * * * *");
   }
}
```

Write a Java method using a *for* loop to display the following:

```
*
**
***
****
```

Hmm, harder. Each line is different but related…
…each line is one asterisk longer…
…so not only do we have a variable number of lines…
…but we have a variable number of things on each line.

Last example our *for* loop printed each row. We will still need that here but now inside the row *for* loop we will need another *for* loop to handle printing the increasing number of asterisks.

Let's put what we know in a table to see what we have…

Write a Java method using a *for* loop to display the following:

```
*
**
***
****
```

Hmm, harder. Each line is different but related…
…each line is one asterisk longer…
…so not only do we have a variable number of lines…
…but we have a variable number of things on each line.

Last example our *for* loop printed each row. We will still need that here but now inside the row *for* loop we will need another *for* loop to handle printing the increasing number of asterisks.

Let's put what we know in a table to see what we have…

Line #	# asterisks
1	
2	
3	
4	

Write a Java method using a *for* loop to display the following:

```
*
**
***
****
```

Hmm, harder. Each line is different but related…
…each line is one asterisk longer…
…so not only do we have a variable number of lines…
…but we have a variable number of things on each line.

Last example our *for* loop printed each row. We will still need that here but now inside the row *for* loop we will need another *for* loop to handle printing the increasing number of asterisks.

Let's put what we know in a table to see what we have…

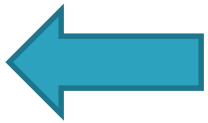
Line #	# asterisks
1	1
2	2
3	3
4	4

Okay cool!
This is easy … the number of asterisks is the same as the line number.

Write a Java method using a *for* loop to display the following:

```
*
**
***
****
```

```
public static void printHalfWedgePointUp() {
    // for each row...
    for ( int i = 1 ; i <= 4 ; i++ ) {
        // the number of asterisks per row
        // is the same as the row number
        for ( int j = 1 ; j <= i ; j++ ) {
            System.out.print("*");
        }
        System.out.println();
    }
```



Line #	# asterisks
1	1
2	2
3	3
4	4

Okay cool!
This is easy … the number of asterisks is the same as the line number.

Hey, why do we need that println() after the inside loop?

Write a Java method using a *for* loop to display the following:

```
****
***
**
*
```

This is similar to the last problem but reversed…
…let's use the same technique as last time…
…put what we know in a table to see how it all relates

How does # asterisks relate to line #?
What is the max number of lines?
…4

Line #	# asterisks
1	4
2	3
3	2
4	1

Write a Java method using a *for* loop to display the following:

```
****
***
**
*
```

This is similar to the last problem but reversed…
…let's use the same technique as last time…
…put what we know in a table to see how it all relates

How does # asterisks relate to line #?
What is the max number of lines?
…4

Line #	# asterisks	Max # lines – line #
1	4	
2	3	
3	2	
4	1	

Write a Java method using a *for* loop to display the following:

```
****
***
**
*
```

This is similar to the last problem but reversed…
…let's use the same technique as last time…
…put what we know in a table to see how it all relates

How does # asterisks relate to line #?
What is the max number of lines?
…4

Line #	# asterisks	Max # lines – line #
1	4	4 – 1 = 3
2	3	4 – 2 = 2
3	2	4 – 3 = 1
4	1	4 – 4 = 0

Now…how does the 3rd column relate to the 2nd column (# *)?

Write a Java method using a *for* loop to display the following:

```
****
***
**
*
```

This is similar to the last problem but reversed…
…let's use the same technique as last time…
…put what we know in a table to see how it all relates

How does # asterisks relate to line #?
What is the max number of lines?
…4

Line #	# asterisks	Max # lines – line #	… + 1
1	4	4 – 1 = 3	4
2	3	4 – 2 = 2	3
3	2	4 – 3 = 1	2
4	1	4 – 4 = 0	1

Now…how does the 3rd column relate to the 2nd column (# *)?

Write a Java method using a *for* loop to display the following:

```
****
***
**
*
```

This is similar to the last problem but reversed…
…let's use the same technique as last time…
…put what we know in a table to see how it all relates

How does # asterisks relate to line #?
What is the max number of lines?
…4

Line #	# asterisks	Max # lines – line #	… + 1
1	4	4 – 1 = 3	4
2	3	4 – 2 = 2	3
3	2	4 – 3 = 1	2
4	1	4 – 4 = 0	1

So the # asterisks is (4 – i + 1) or (3 – i)!

Write a Java method using a *for* loop to display the following:

```
****
***
**
*
```

```
public static void printHalfWedgePointDown() {
   // for each row...
   for ( int i = 1 ; i <= 4 ; i++ ) {
      // the number of asterisks is the max # lines
      // minus the line number + 1
      for ( int j = 3 - i ; j >= 1 ; j-- ) {
         System.out.print("*");
      }
      System.out.println();
   }
}
```

Line #	# asterisks	Max # lines – line #	… + 1
1	4	4 – 1 = 3	4
2	3	4 – 2 = 2	3
3	2	4 – 3 = 1	2
4	1	4 – 4 = 0	1

So the # asterisks is (4 – i + 1) or (3 – i)!

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…

…we now have 2 characters per line…

…leading spaces following by training asterisks

Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?
2) How does # asterisks relate to line #?

Line #	# " "	# "*"
1	3	1
2	2	3
3	1	5
4	0	7

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…
…we now have 2 characters per line…
…leading spaces following by training asterisks

Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?

2) How does # asterisks relate to line #?

Line #	# " "		# "*"	
1	3		1	
2	2		3	
3	1		5	
4	0		7	

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…
…we now have 2 characters per line…
…leading spaces following by training asterisks

Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?

2) How does # asterisks relate to line #?

Line #	# " "		# "*"	
1	3		1	
2	2		3	
3	1		5	
4	0		7	

Max # lines is 4

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…
…we now have 2 characters per line…
…leading spaces following by training asterisks

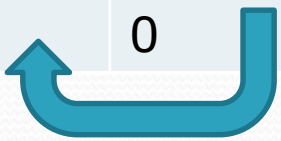
Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?

2) How does # asterisks relate to line #?

Line #	# " "	4 – line #	# "*"	
1	3	3	1	
2	2	2	3	
3	1	1	5	
4	0	0	7	



Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…
…we now have 2 characters per line…
…leading spaces following by training asterisks

Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?

2) How does # asterisks relate to line #?

Line #	# " "	4 – line #	# "*"
1	3	3	1
2	2	2	3
3	1	1	5
4	0	0	7

"*" are the odd #'s

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…
…we now have 2 characters per line…
…leading spaces following by training asterisks

Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?

2) How does # asterisks relate to line #?

Line #	# " "	4 – line #	# "*"	2(line #) - 1
1	3	3	1	1
2	2	2	3	3
3	1	1	5	5
4	0	0	7	7

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

This is different because…
…we now have 2 characters per line…
…leading spaces following by training asterisks

Again, let's put what we know in a table…

Two questions:

1) How does # spaces relate to line #?

2) How does # asterisks relate to line #?

Line #	# " "	4 – line #	# "*"	2(line #) - 1
1	3	3	1	1
2	2	2		
3	1	1		
4	0	0	7	7

Inside the row loop we have 2 things varying, one after the other…

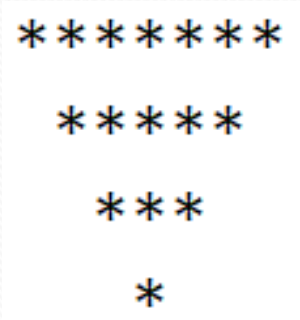
…that means we will need two separate loops inside the row loop.

Write a Java method using a *for* loop to display the following:

```
   *
  ***
 *****
*******
```

```
public static void printWedgePointUp() {
    // for each row...
    for ( int i = 1 ; i <= 4 ; i++ ){
        // print leading spaces
        for ( int j = 1 ; j <= 4 - i ; j++ ) {
            System.out.print(" ");
        }
        // print training astericks
        for ( int j = 1 ; j <= 2*i - 1 ; j++ ) {
            System.out.print("*");
        }
        // end the line
        System.out.println();
    }
}
```

Homework:



Write a Java method using a *for* loop to display this figure.