Nested Loops

Just as if statements can be nested, so can loops. You can place a while loop within a while loop, a for loop within a for loop, a while loop within a for loop, or any other combination you can think of.

The following examples and exercises are designed to develop and practice using nested loops.

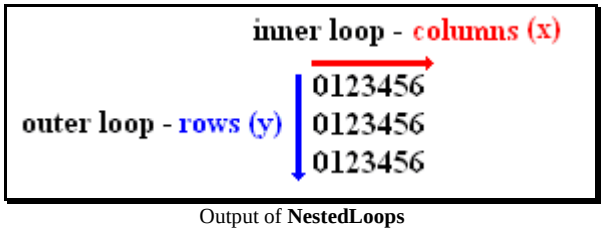
These programs require no input and since the processing and output are combined you may write the entire program within the class constructor as was done in the example program below.

The example program shown below displays **0123456** three times.

```
public class NestedLoops
{
    public static void main(String args[]) {
        NestedLoops loops = new NestedLoops();
    }

    public NestedLoops( ) {
        output();
    }

    public void output() {
        for (int y = 0; y < 3; y++)          // executes 3 times controlling the number of rows
        {
            for(int x = 0; x < 7; x++)       // executes 7 times controlling the number of cols
                System.out.print(x);         // prints the values 0 -> 6.  x begins at 0
            System.out.println();            //                            and terminates at 6
        }
    }
}
```



Output of **NestedLoops**

Lab 16 - ASSIGNMENT

Lab 16A - 60 points

OBJECTIVE

WAP that reads two integers from the keyboard representing the *width* and *height* of a rectangle. Output a rectangle of stars corresponding to the *width* and *height* of the rectangle.

Hint: The outer loop controls the number of rows. The inner loop controls the number of columns.

FIELD SUMMARY

- **int width** – the width of the rectangle.
- **int height** – the height of the rectangle.

METHOD SUMMARY

- **main** – instantiate an object of your class. Make method calls to input, and output.
- **input** – declare a `Scanner` object and read the *width* and *height* of a rectangle from the keyboard using an appropriate prompt.
- **output** – Use a nested loop to output a *width* by *height* rectangle of stars.

SAMPLE KEYBOARD INPUT

Enter the width and height of the rectangle: 6 5

SAMPLE OUTPUT

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

Lab 16B - 70 points

OBJECTIVE

WAP to read the *height* of a triangle from the keyboard. Output a triangle using the design shown below.

Hint: The *terminating value* of the inner loop is the loop control variable of the outer loop.

FIELD SUMMARY

- **int height** – the height of the triangle.

METHOD SUMMARY

- **main** – instantiate an object of your class. Make method calls to input, and output.
- **input** – declare a `Scanner` object and read the *height* of a triangle from the keyboard using an appropriate prompt.
- **output** – Use a nested loop to output a triangle *height* tall as shown below..

SAMPLE KEYBOARD INPUT

Enter the height of the triangle: 5

SAMPLE OUTPUT

1
12
123
1234
12345

Lab 16C - 80 points

OBJECTIVE

WAP that read an integer from the keyboard. Output a descending sequence of numbers that begins with the number read from the keyboard and decreased down to 4. Then, on a new line, beginning with the number read from the keyboard minus 1, repeat this pattern. Continue repeating the pattern until the sequence "6 5 4" is output. See the examples below.

Hint: Look at where the outer loop starts and begins. Look at where the inner loop starts and begins. All values are decreasing therefore both loops are counting backwards.

FIELD SUMMARY

- **int num** – the number read from the keyboard used to determine the shape of the pattern..

METHOD SUMMARY

- **main** – instantiate an object of your class. Make method calls to input, and output.
- **input** – declare a `Scanner` object and read an integer from the keyboard using an appropriate prompt. The value entered will always be greater than 5.
- **output** – Use a nested loop to output sequence of numbers as shown below.

SAMPLE KEYBOARD INPUT

Enter a number: **9**

SAMPLE OUTPUT

9 8 7 6 5 4
8 7 6 5 4
7 6 5 4
6 5 4

SAMPLE KEYBOARD INPUT

Enter a number: **12**

SAMPLE OUTPUT

12 11 10 9 8 7 6 5 4
11 10 9 8 7 6 5 4
10 9 8 7 6 5 4
9 8 7 6 5 4
8 7 6 5 4
7 6 5 4
6 5 4

Lab 16D - 90 points

OBJECTIVE

WAP that reads a number from the keyboard that represents the *height* of a triangle of stars. Output a corresponding triangle of stars.

Hint: The concept is the same as Lab16C. The difference is in the print statement. Print one *.

SAMPLE KEYBOARD INPUT

Enter the height of the triangle: **7**

SAMPLE OUTPUT

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

Lab 16E - 100 points

OBJECTIVE

Write a program that reads a number from the keyboard and draws a rocket ship on the screen. The size of the rocket is determined by the input value. Before you start trying to code this project study the rocket. How does the number entered from the keyboard affect the shape? Look for patterns. Plan your solution before you start the project.

Hint: Use lots of methods. For example you can write a method that will output a specified number of blank spaces, i.e.:

```
/* This method prints n number of spaces */
public void spaces(int n) {
    for (int i = 0; i < n; i++)
        System.out.print(" ");
}
```

Look for similarities. For example, the nose and the base are the same. One method can print them both. The *height* is the key to the entire program. Look at the relationship between *height* and various parts of the spaceship.

SAMPLE KEYBOARD INPUT

Enter the height of the rocket ship: **2**

SAMPLE OUTPUT

```
    /\
   /**\
  //**\\
 ///**\\\
+=*=*=*=*+
|./\../\.|
|/\/\/\/\|
|\/\/\/\/|
|./\../\.|
+=*=*=*=*+
|\/\/\/\/|
|./\../\.|
|/\/\/\/\|
+=*=*=*=*+
   /**\
  //**\\
 ///**\\\
```

SAMPLE KEYBOARD INPUT

Enter the height of the rocket ship: **5**

SAMPLE OUTPUT