

Nested loops worksheet

Arjun Chandrasekhar

Getting started

In this worksheet you will complete some practice problems related to your reading on nested loops. Here is how you should complete the worksheet:

Create a file in your text editor (either Notepad++ or BBedit). Put the following starter code into it:

```
public class NestedLoops
{
    public static void main(String[] args)
    {

    }
}
```

Save your file as `NestedLoops.java`.

Open up the command line application (either Terminal or Command Prompt). Navigate to the location where you saved the file.

How to complete the worksheet

For each problem, do the following:

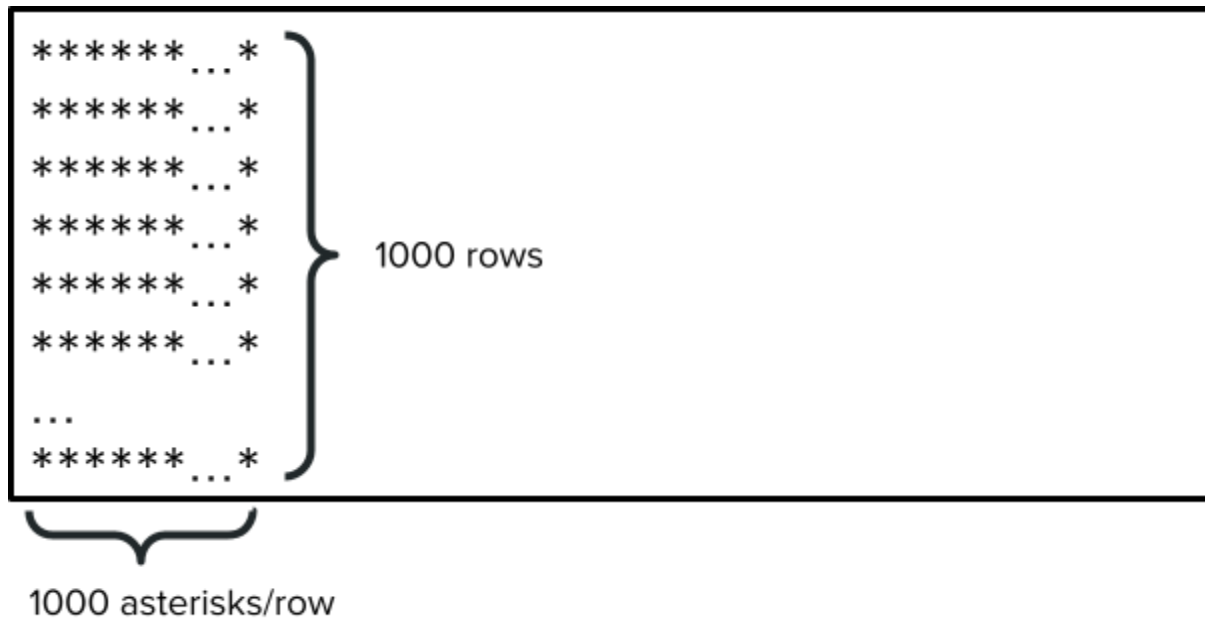
1. Answer the problem to the best of your abilities.
2. Put the code into `NestedLoops.java`, run the code, and see what happens!
 - a. First, compile the code: `javac NestedLoops.java`
 - b. Next, run the code: `java NestedLoops`
3. If your prediction was wrong, then discuss amongst yourselves - or perhaps find another group to ask for help!

Miscellaneous

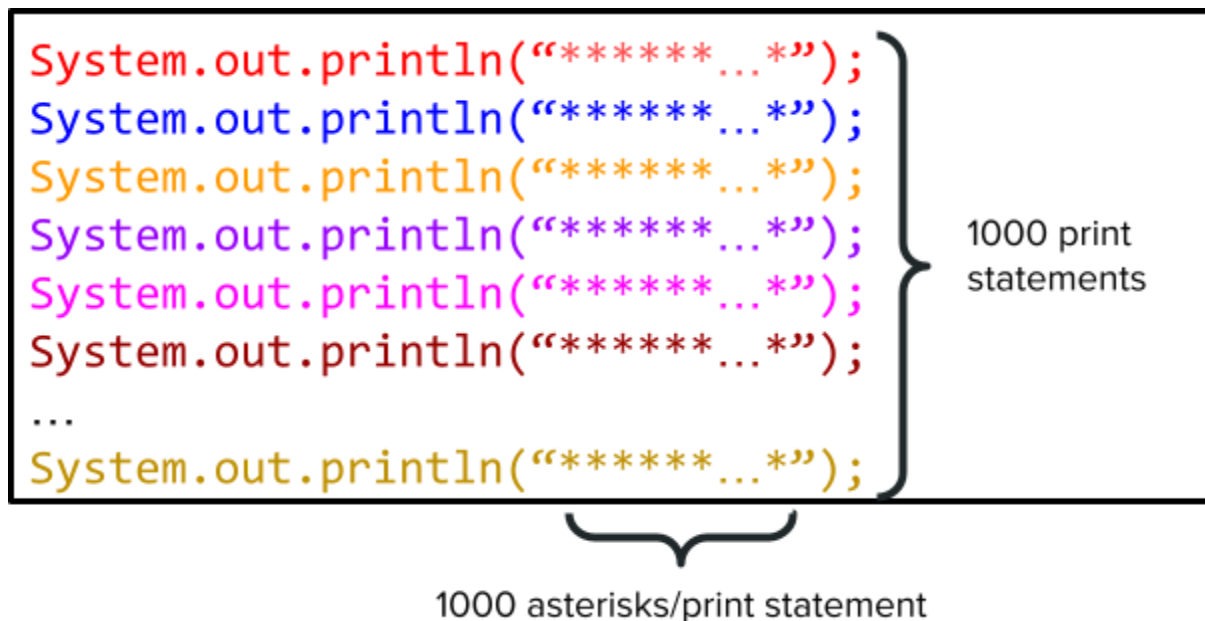
- Have one person dictate the code to write, and have another person type the code. Over the course of the semester, make sure you spend plenty of time in both roles.
- Some problems ask you to trace/walk through the code step by step. It may help to manually add line numbers.

Repetitive loops

Suppose we want to print out a 1000x1000 square of asterisks

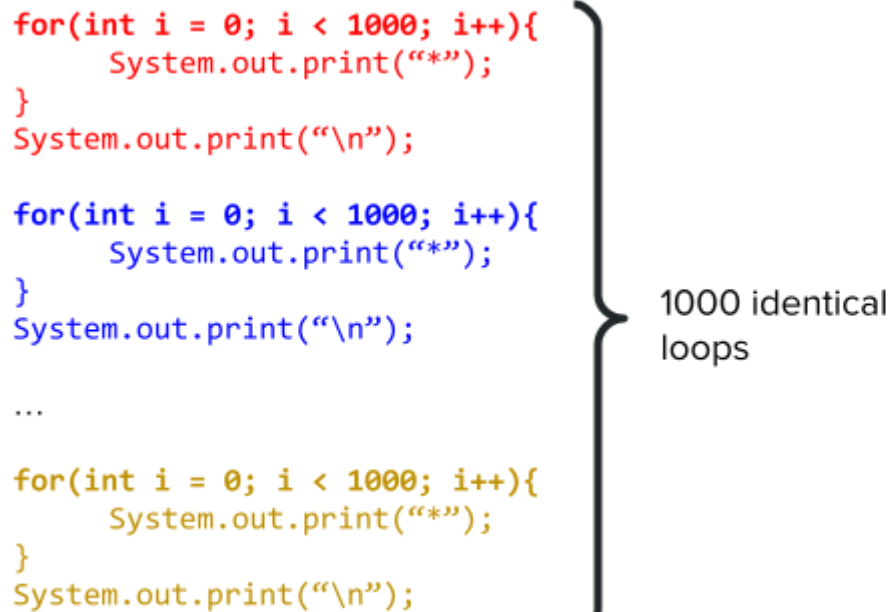


What are some of the issues with the following approach?



How might loops be able to help us here?

OK, let's try one approach involving loops. Examine the following code:



```
for(int i = 0; i < 1000; i++){  
    System.out.print("*");  
}  
System.out.print("\n");  
  
for(int i = 0; i < 1000; i++){  
    System.out.print("*");  
}  
System.out.print("\n");  
  
...  
  
for(int i = 0; i < 1000; i++){  
    System.out.print("*");  
}  
System.out.print("\n");
```

1000 identical loops

Explain what advantages and disadvantages this approach has compared to the previous approach, and explain what are the fundamental issues that both approaches share.

Constructing nested loops

Write code to produce the following output using nine print statements.

10*2=20

10*4=40

10*6=60

20*2=40

20*4=80

20*6=120

30*2=60

30*4=120

30*6=180

Run your code to confirm that it is correct.

Did you use copy-paste to create the code? Did you ever have to make any changes after copy-pasting? With this in mind, discuss two sources of repetition in this code.

Write code to produce the following output using three for loops. The three colors indicate which 3 chunks of code should be re-written as a loop.

You will need to use String concatenation involving your loop variable!

10*2=20

10*4=40

10*6=60

20*2=40

20*4=80

20*6=120

30*2=60

30*4=120

30*6=180

Run your code to confirm that it is correct.

Compared to the previous approach, how does this approach make the code less repetitive? What sources of repetition are still present?

Write code to produce the following output using nested loops.

You will need to use String concatenation involving your two loop variables!

10*2=20

10*4=40

10*6=60

20*2=40

20*4=80

20*6=120

30*2=60

30*4=120

30*6=180

Run your code to confirm that it is correct.

Structure correlates with function!

In this class there are a few overarching principles that I want you to be aware of. One of them is the following:

The structure of the data
Dictates
The structure of the code

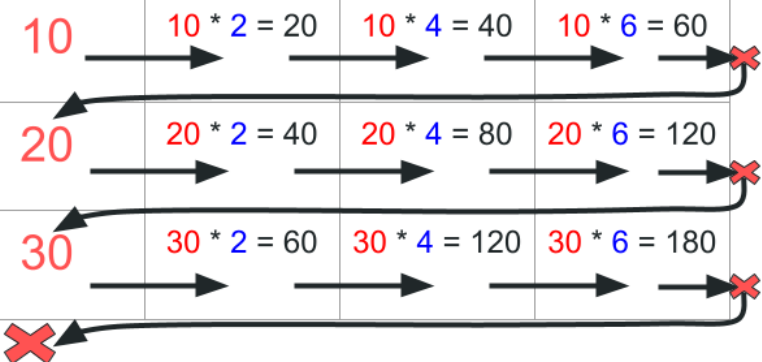
Nested loops are our first example of this idea.

Nested Loops

10*2=20
10*4=40
10*6=60
20*2=40
20*4=80
20*6=120
30*2=60
30*4=120
30*6=180

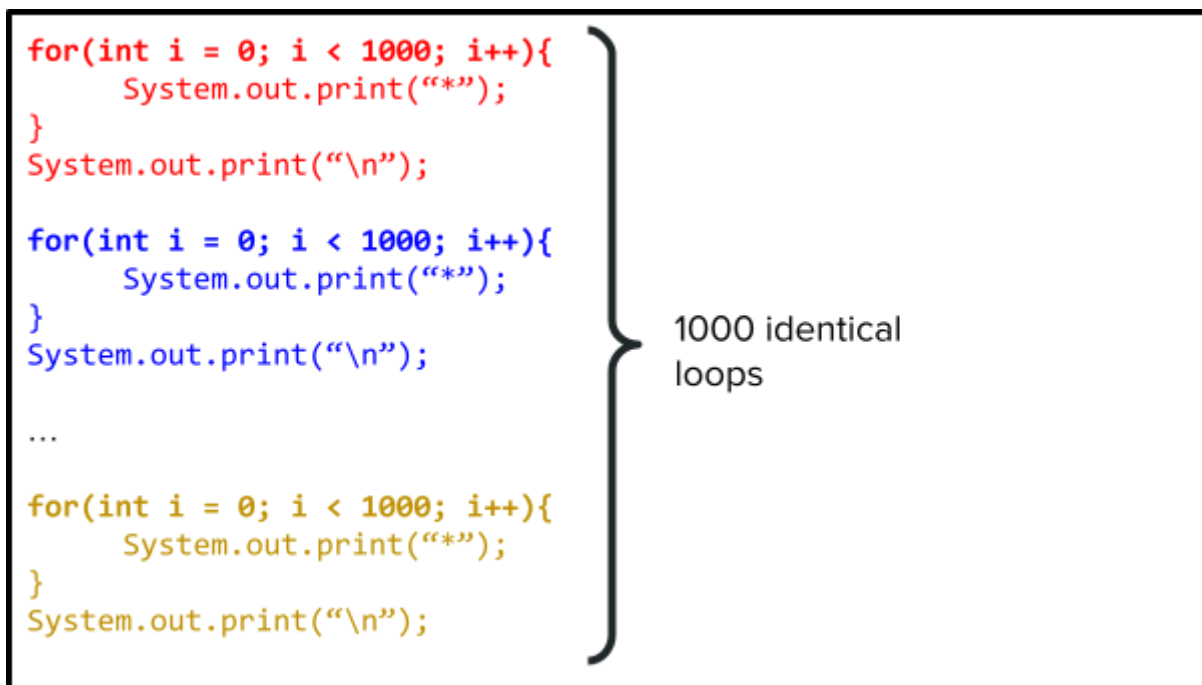
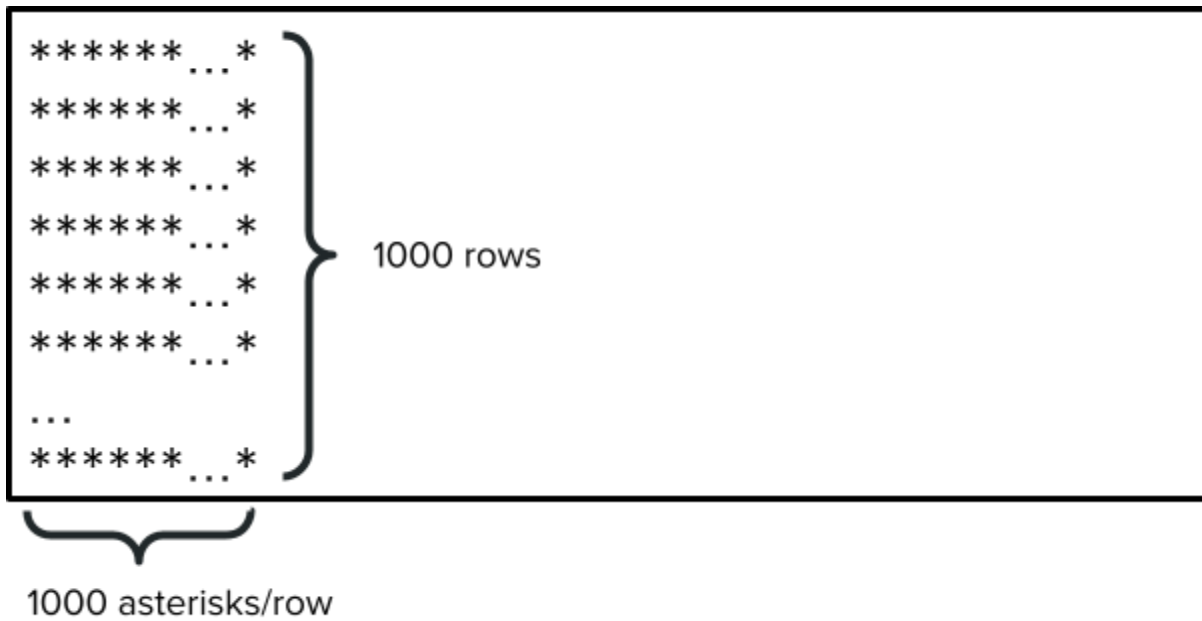
2-D data suggests 2-D
code structure...

	2	4	6
10	10 * 2 = 20	10 * 4 = 40	10 * 6 = 60
20	20 * 2 = 40	20 * 4 = 80	20 * 6 = 120
30	30 * 2 = 60	30 * 4 = 120	30 * 6 = 180



There will be project report and exam questions that ask you why you chose to use nested loops. You should be prepared to explain why 2-D data dictates using 2-D looping.

Let's return to the original problem, and the flawed solution that we examined:



Explain why the structure of our output data dictates the use of nested loops. Then, brainstorm how we might use nested loops to fix our flawed solution.

Let's examine some sample output that your Wordle project should be able to generate:

```
Welcome to Wordle!
Try to guess the 5-letter word in 6 guesses!

Make your next guess: crane
                     _***_

Make your next guess: stark
                     _**_

Make your next guess: train
                     _**_*

Make your next guess: range
                     *$$_

Make your next guess: waner
                     -$$-$

Make your next guess: manor
                     $$$$$

Congrats! You guessed the word!
```

Explain why the structure of the output dictates the use of nested loops.

- **What are the “dimensions” of our output? What two dimensions of looping do we need?**

Heterogeneous nested loops

Suppose we want to print out a triangular pattern of asterisks from 1 to 1000

```
*  
**  
***  
****  
*****  
*****  
...  
***** ... *
```



1000 asterisks

What is the dimensionality of our output data? What does that tell us about how our code should be structured to solve this problem?

Consider the following two approaches to print out a triangle of asterisks:

```
System.out.println("*");
System.out.println("**");
System.out.println("***");
System.out.println("****");
System.out.println("*****");
System.out.println("*****");
...
System.out.println("*****...*");
```



1000 asterisks

```
for(int i = 0; i < 1; i++){
    System.out.print("*");
}
System.out.print("\n");

for(int i = 0; i < 2; i++){
    System.out.print("*");
}
System.out.print("\n");

...

for(int i = 0; i < 1000; i++){
    System.out.print("*");
}
System.out.print("\n");
```

Discuss the pitfalls and limitations of these approaches, and brainstorm how we might use nested loops to solve this problem. If you are not sure how to solve this with nested loops, feel free to move ahead in the worksheet and come back to this problem.

Evaluating heterogeneous nested loops

For each of the following problems, evaluate how the code will run to answer the question about the code. For each problem, do the following:

1. Work through the problem and write down your prediction.
2. Run the code, and check if your prediction was correct.
3. If your prediction was wrong, use either or both of the following approaches to see where you went wrong:
 - a. Manually trace the code. Write down each line number/instruction that gets executed, in the order that it gets executed, and state what effect that instruction will have (including/especially how the values of the variables change).
 - b. Add print statements to your code to make the code show the progression of the values of the relevant variables.
 - i. To make your output more readable, you can use String concatenation, i.e. `System.out.println("x = " + x);`

What will be the value of `x` after running this code?

```
int n = 4;
int x = 0;
while(n > 0) {
    for(int i = 0; i < n; i++) {
        x++;
    }
    n /= 2;
}
```

What will be the value of `x` after running this code?

```
int x = 0;
for(int i = 1; i <= 3; i++) {
    for(int j = 1; j <= i; j++) {
        x += j;
    }
}
```

What will be the output of this code?

What is the effect/purpose of having the command `System.out.print("\n")` located where it is?

```
String name = "ABC";
for(int i = 0; i < name.length(); i++) {
    for(int j = 0; j <= i; j++) {
        System.out.print(name.charAt(j));
    }
    System.out.print("\n");
}
```

What will be the value of `str` after running this code?

```
String name = "Arjun";
String str = "";
for(int i = 0; i < name.length(); i += 2) {
    for(int j = i; j < name.length(); j += 2) {
        str += name.charAt(j);
    }
}
```

What will be the value of `str` after running this code?

```
String name = "Arjun";
String str = "";
for(int i = 0; i < name.length(); i += 2) {
    for(int j = i; j < name.length(); j++) {
        str += name.charAt(j);
    }
}
```

What will be the value of `str` after running this code?

```
String name = "Arjun";
String str = "";
for(int i = 0; i < name.length(); i++) {
    for(int j = i; j < name.length(); j += 2) {
        str += name.charAt(j);
    }
}
```

What will be the value of `str` after running this code?

```
String name = "Arjun";
String str = "";
for(int i = 0; i < name.length(); i++) {
    for(int j = i; j < name.length(); j++) {
        str += name.charAt(j);
    }
}
```