```
/******************************************************************************\
|************************ For Loops of Varying Size ***************************|
|******************************************************************************|

Consider that we would like to print the following pattern for a given value of
SIZE, where 3 < SIZE < 9:

SIZE 3:             SIZE 6:
////////\\\\\\\\    ////////////////\\\\\\\\\\\\\\\\
////********\\\\    ////////////********\\\\\\\\\\\\
****************    ////////****************\\\\\\\\
                    ////************************\\\\
                    ********************************

Step 1: Determine the different patterns associated with the problem.
This will indicate the number of for loops that we need to have in our solution.
Ask yourself: "How many different symbols are incorporated into the problem?"

When creating patterns, it's a good idea to separate the problem line by line.
When doing this, create an outer for loop to manage the number of lines that we
would like to print out.

//       +-----------------------------------+
//       | This part of the for loop tells   |
//       | the program to print three lines. |
//       | This number will change depending |
//       | on the number of lines we would   |
//       | like to print out, which will     |
//       | vary depending on the SIZE that   |
//       | we are printing out. For now, we  |
//       | can leave this at three, however, |
//       | we will need to change it later   |
//       | so that it prints the correct     |
//       | number of lines based on the SIZE |
//       | of the figure we have chosen.     |
//       +-----------------------------------+
//                        |
//                        |
//                       \|/
//                        V
for(int line = 1; line <= 3; line++)
{
    // OTHER CODE WILL GO HERE

    // Including this println on the end will
    // allow the program to move to the next line.
    System.out.println();
}

We will then need a for loop for each symbol, "/", "*" and "\",
to control the number of times each character is printed per line.
These for loops will go inside of the line number for loop, like so:

// Line
for(int line = 1; line <= 3; line++)
{
    // Print each "/"
    for()
    {
    System.out.println("/");
    }

    // Print each "*"
    for()
    {
        System.out.println("*");
    }
    // Print each "\"
    for()
    {
        System.out.println("\\");
    }
}
```

Step 2: Construct a table for at least one of the SIZES.
Let's use SIZE 3. SIZE 6 is included for reference as well.

SIZE = 3			
line	/	*	\
1	8	0	8
2	4	8	4
3	0	16	0

SIZE = 6			
line	/	*	\
1	24	0	24
2	16	8	16
3	12	16	12
4	8	24	8
5	4	32	4
6	0	40	0

Step 3: Determine the mathematical expressions
Each character in the table is going to need it's own expression.
This will tell the computer how many times it needs to print that character for the line number that it's currently at.

For this reason, we need an expression dependent upon the line number.

Let's start with the slashes.
We need to find an equation that satisfies the following requirements:

When line is 1, the # of slashes = 8
When line is 2, the # of slashes = 4
When line is 3, the # of slashes = 0

line	Expression	/
1	line * _______	= 8
2	line * _______	= 4
3	line * _______	= 0

A possible solution: 12 - (4*line)

Check if the expression satisfies the conditions:

line	Expression	/	
1	12 - (4*line) =	8	-> 12 - (4*1) = 8
2	12 - (4*line) =	4	-> 12 - (4*2) = 4
3	12 - (4*line) =	0	-> 12 - (4*3) = 0

Therefore, the expression, 12 - (4*line), gives us the number of slashes that we need to print out for each line.

Set up the for loop accordingly:

```
// Print each "/"
for(int f = 1; f <= 12 - (4*line); f++)
{
    System.out.print("/");
}
```

Now for the stars.
We need to find an equation that satisfies the following requirements:

```

When line is 1, the # of stars = 0
When line is 2, the # of stars = 8
When line is 3, the # of start = 16
+---------------------------+
|  line    Expression    *  |
|---------------------------|
|  1     line * ______  = 0  |
|  2     line * ______  = 8  |
|  3     line * ______  = 16 |
+---------------------------+
A possible solution: 8 * (line-1)

Check if the expression satisfies the conditions:
+---------------------------+
|  line   Expression     *  |
|---------------------------+
|  1      8 * (line-1) = 8  | -> 8 * (1-1)  = 0
|  2      8 * (line-1) = 4  | -> 8 * (2-1)  = 8
|  3      8 * (line-1) = 0  | -> 8 * (3-1)  = 16
+---------------------------+

Therefore, the expression, 12 - (4*line), gives us the
number of "/" that we need to print out for each line.

Set up the for loop accordingly:

// Print each "/"
// +------------------------------------+
// | This part of the for loop will     |
// | change depending on the SIZE that  |
// | we are working on. In this case,   |
// | Starting with 12 works when SIZE   |
// | is equal to 3; however, this will |
// | will not work for any other SIZE,  |
// | and so we need to change this.     |
// +------------------------------------+
//                   |
//                   |
//                  \|/
//                   V
for(int f = 1; f <= 12 - (4*line); f++)
{
    System.out.print("/");
}


Incorporating SIZE:
Since the SIZE will end up changing the number of slashes that we will be
starting with, we need to include it into our formulas above for slashes.

Consider the tables again. Notice how we start with 8 slashes in SIZE 3, while
we start with 16 in SIZE 6:
+------------+       +------------+
|  SIZE = 3  |       |  SIZE = 6  |
|============+       |============|
|  line   /  |       |  line   /  |
|------------|       |------------|
|  1      8  |       |  1      20 |
|  2      4  |       |  2      16 |
|  3      0  |       |  3      12 |
+------------+       |  4      8  |
                     |  5      4  |
                     |  6      0  |
                     +------------+

We need to change our initial starting number in the slashes equation:

```

```
When SIZE is 3: We start with 12 slashes
When SIZE is 6: We start with 20 slashes
+---------------------------+
|  SIZE    Expression   /   |
|---------------------------|
|  3     SIZE * _____ = 12  |
|  6     SIZE * _____ = 20  |
+---------------------------+
Possible Solution: SIZE*4

This will change the slashes equation to as follows:
Originally:      12    - (4*line) <-- This only works for SIZE = 3
New Equation: (SIZE*4) - (4*line) <-- This works for all SIZES
\*******************************************************************************/
public class StarFigure
{
    // SIZE is a constant int
    public static final int SIZE = 6;

    public static void main(String[] args)
    {
        // Print out each line
        for(int line = 1; line <= SIZE; line++)
        {
            // Within each line...
            // Print out each forward slash
            for(int f = 1; f <= (4*SIZE)-(4*line); f++)
            {
                System.out.print("/");
            }

            // Print out each star
            for(int s = 1; s <= 8*(line-1); s++)
            {
                System.out.print("*");
            }

            // Print out each backslash
            for(int b = 1; b <= (4*SIZE)-(4*line); b++)
            {
                System.out.print("\\");
            }

            // Drop down to the next line
            System.out.println();
        }
    }
}
```