

Pattern Printing (1–9)

Loops making art. The classic beginner challenge. ■

■ What You Will Learn

- How nested loops create 2D patterns
- Row = outer loop, Column = inner loop
- Right triangle, inverted triangle, pyramid patterns
- Number and star variations

■ Intuition First

Patterns are just **loops within loops**. The outer loop controls the *rows* (how many lines). The inner loop controls the *columns* (how many things per line). Once you see this, every pattern becomes obvious.

■ Think Like This

Think of printing a grid. For each row (outer loop), you print some symbols (inner loop), then move to the next line. The trick is figuring out HOW MANY symbols to print per row — that's the pattern logic.

■ Pattern Gallery

Pattern 1: Right Triangle Pattern 2: Inverted

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

Pattern 3: Pyramid Pattern 4: Numbers

```
* 1
* * 1 2
* * * 1 2 3
* * * * 1 2 3 4
* * * * * 1 2 3 4 5
```

■■ How to Think About Any Pattern

1	Count the rows How many lines? That's your outer loop: <code>for(i=1; i<=n; i++)</code>
2	Count items per row In row <code>i</code> , how many items print? Often <code>= i</code> (for triangle).
3	Handle spaces For pyramid: print <code>(n-i)</code> spaces, then <code>i</code> stars.
4	Inner loop prints items <code>for(j=1; j<=i; j++) System.out.print("* ");</code>
5	Newline after each row <code>System.out.println();</code> at end of outer loop body.

■ Code — 3 Key Patterns

```
// Pattern 1: Right Triangle
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print("* ");
    }
    System.out.println();
}

// Pattern 2: Inverted Triangle
for (int i = 5; i >= 1; i--) {
    for (int j = 1; j <= i; j++) {
        System.out.print("* ");
    }
    System.out.println();
}

// Pattern 3: Pyramid (centred)
int n = 5;
for (int i = 1; i <= n; i++) {
    for (int s = 1; s <= n-i; s++) System.out.print(" "); // spaces
    for (int j = 1; j <= i; j++) System.out.print("* "); // stars
    System.out.println();
}
```

■■ Time & Space Complexity

■ TIME COMPLEXITY

$O(n^2)$

■ SPACE COMPLEXITY

$O(1)$

Two nested loops each running up to `n` $\rightarrow$ `n` $\times$ `n` operations = $O(n^2)$. No extra memory needed.

Common Mistakes

print vs println inside inner loop

Use `System.out.print()` (no newline) inside the inner loop. Use `System.out.println()` AFTER the inner loop to end each row.

Practice Problems

#	Problem	Difficulty
1	Print a right triangle of stars for n rows	Easy
2	Print an inverted right triangle	Easy
3	Print a number triangle (1, 12, 123...)	Easy
4	Print a hollow square border of stars	Medium
5	Print a diamond pattern (up then down)	Medium

Quick Revision Box

- ✓ Outer loop = rows
- ✓ Inner loop = columns (items per row)
- ✓ Use `print()` inside inner loop, `println()` after
- ✓ Space management = key to pyramid/diamond shapes
- ✓ `i` controls row count, `j` controls column count
- ✓ Pattern problems test nested loop mastery

Visual Thinking

Imagine It Like This

Think of it like building with LEGO bricks on a grid. Each outer loop step = new row. Each inner loop step = placing one brick. Animation: Show a grid where cells light up row by row, column by column — a wave of stars filling a triangle.