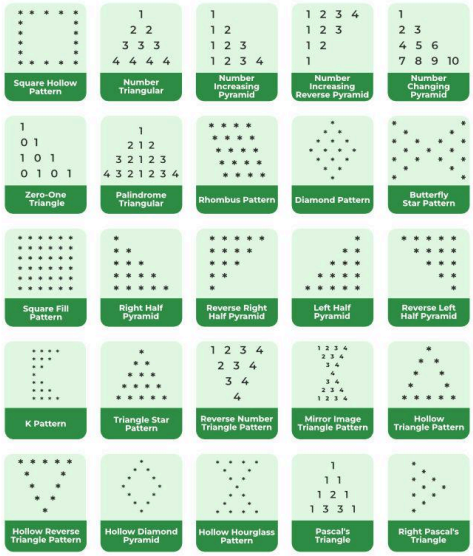
Java Pattern Programs - Learn How to Print Pattern in Java

Last Updated : 08 Apr, 2025

In many Java interviews **Star, number,** and **character patterns** are the most asked **Java Pattern Programs** to check your logical and coding skills. **Pattern programs in Java** help you to sharpen your looping concepts (especially for loop) and problem-solving skills in Java. If you are looking for a place to get all the **Java pattern programs with solutions**, stop your search here.

Here, we have compiled a **top pattern exercises on Java.**

Prerequisite: Remember that to learn pattern programs, you must know Java Loops (for, while, do-while) and basic syntax.



Patterns Programs in Java

Java Pattern Programs

Here, you will find the top 25 Java pattern programs with their proper code and explanation.

Table of Content

All Pattern Programs in Java are mentioned below:

1. Square Hollow Pattern

This program prints a square where the border is filled with stars (*), and the inside is hollow (filled with spaces).

```
// Java Program to print pattern
```

```java
// Square hollow pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle number of rows
        for (i = 0; i < n; i++) {

            //  inner loop to handle number of columns
            for (j = 0; j < n; j++) {

                // star will print only when  it is in first
                // row or last row or first column or last
                // column
                if (i == 0 || j == 0 || i == n - 1
                    || j == n - 1) {
                    System.out.print("*");
                }

                // otherwise print space only
                else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

Related searches

Java Pattern Programs Pdf | How to Write Pattern Program in Java

2. Number Triangle Pattern

Prints a right-angled triangle with numbers in increasing row order, aligned to the right.

```
// Java Program to print pattern
// Number triangle pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;
```

```
        // outer loop to handle number of rows
        for (i = 1; i <= n; i++) {

            // inner loop to print space
            for (j = 1; j <= n - i; j++) {
                System.out.print(" ");
            }

            // inner loop to print star
            for (j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }

            // print new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
     1
    2 2
   3 3 3
  4 4 4 4
 5 5 5 5 5
6 6 6 6 6 6
```

3. Number-Increasing Pyramid Pattern

Prints a pyramid where each row contains numbers from 1 to the row number.

```java
// Java Program to print pattern
// Number-increasing pyramid
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle number of rows
        for (i = 1; i <= n; i++) {

            // inner loop to handle number of columns
            for (j = 1; j <= i; j++) {

                // printing column values
                // upto the row value
                System.out.print(j + " ");
            }

            // print new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
1
1 2
1 2 3
```

```
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

4. Number-Increasing Reverse Pyramid Pattern

This is a reverse pyramid where each row starts from 1 and ends at the row count, in decreasing number of elements.

```java
// Java Program to print pattern
// Number-increasing reverse pyramid
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle number of rows
        for (i = n; i >= 1; i--) {

            // inner loop to handle number of columns
            for (j = 1; j <= i; j++) {

                // printing column values
                // upto the row value
                System.out.print(j + " ");
            }

            // print new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
```

```
        printPattern(n);
    }
}
```

Output

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

5. Number-Changing Pyramid Pattern

Prints a pyramid where numbers increase continuously from top to bottom.

```
// Java Program to print pattern
// Number-changing pyramid
import java.util.*;

// Java code for printing pattern
public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;
        int num = 1;

        // outer loop to handle number of rows
        for (i = 1; i <= n; i++) {

            // inner loop to handle number of columns
            for (j = 1; j <= i; j++) {

                // printing value of num in each iteration
                System.out.print(num + " ");
```

```
                // increasing the value of num
                num++;
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

6. Zero-One Triangle Pattern

Prints a triangle where each number alternates between 1 and 0 based on the position.

```
// Java Program to print pattern
// Zero-One triangle
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
```

```
    {
        int i, j;

        //outer loop to handle number of rows
        for (i = 1; i <= n; i++) {

            //inner loop to handle number of columns
            for (j = 1; j <= i; j++) {

                // if the sum of (i+j) is even then print 1
                if ((i + j) % 2 == 0) {
                    System.out.print(1 + " ");
                }
                // otherwise print 0
                else {
                    System.out.print(0 + " ");
                }
            }

            //printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
```

7. Palindrome Triangle Pattern

Prints a triangle with mirrored numbers forming a palindrome on each row.

```java
// Java Program to print pattern
// Palindrome triangle
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle number of rows
        for (i = 1; i <= n; i++) {

            // inner loop to print the spaces
            for (j = 1; j <= 2 * (n - i); j++) {
                System.out.print(" ");
            }

            // inner loop to print the first part
            for (j = i; j >= 1; j--) {
                System.out.print(j + " ");
            }

            // inner loop to print the second part
            for (j = 2; j <= i; j++) {
                System.out.print(j + " ");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
```

```
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
          1
        2 1 2
      3 2 1 2 3
    4 3 2 1 2 3 4
  5 4 3 2 1 2 3 4 5
6 5 4 3 2 1 2 3 4 5 6
```

8. Rhombus Pattern

Prints a rhombus (tilted square) made of stars, shifted by spaces.

```
// Java Program to print
// Rhombus pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;
        int num = 1;

        // outer loop to handle number of rows
        for (i = 1; i <= n; i++) {

            // inner loop to print spaces
            for (j = 1; j <= n - i; j++) {
                System.out.print(" ");
            }
```

```
            // inner loop to print stars
            for (j = 1; j <= n; j++) {
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

9. Diamond Star Pattern

Prints a diamond shape made of stars.

```
// Java Program to print
// Diamond Star Pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
```

```
{
    int i, j;
    int num = 1;

    // outer loop to handle upper part
    for (i = 1; i <= n; i++) {

        // inner loop to print spaces
        for (j = 1; j <= n - i; j++) {
            System.out.print(" ");
        }

        // inner loop to print stars
        for (j = 1; j <= 2 * i - 1; j++) {
            System.out.print("*");
        }
        System.out.println();
    }

    // outer loop to handle lower part
    for (i = n-1; i >= 1; i--) {

        // inner loop to print spaces
        for (j = 1; j <= n - i; j++) {
            System.out.print(" ");
        }

        // inner loop to print stars
        for (j = 1; j <= 2 * i - 1; j++) {
            System.out.print("*");
        }
        System.out.println();
    }
}

// Driver Function
public static void main(String args[])
{
    int n = 6;
    printPattern(n);
}
```

```
}
```

Output

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

10. Butterfly Star Pattern

Prints a butterfly-shaped pattern using stars (*).

```
// Java Program to print Butterfly Pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate the pattern
    public static void printPattern(int n)
    {
        // Outer loop to handle the upper part
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= 2 * n; j++) {

                // To print spaces
                if (j > i && j <= 2 * n - i) {
                    System.out.print(" ");
                }

                // To print stars
                else {
```

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

        // Outer loop to handle the lower part
        for (int i = n; i >= 1; i--) {
            for (int j = 1; j <= 2 * n; j++) {

                // To print spaces
                if (j > i && j <= 2 * n - i) {
                    System.out.print(" ");
                }

                // To print stars
                else {
                    System.out.print("*");
                }
            }
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String[] args)
    {
        // Number of rows
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

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

11. Square Fill Pattern

This program prints a filled square of stars (*) with n+1 rows and columns.

```java
// Java Program to print
// Square fill pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = 0; i < n; i++) {

            // inner loop to handle columns
            for (j = 0; j < n; j++) {
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
```

```
        printPattern(n);
    }
}
```

Output

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

12. Right Half Pyramid Pattern

This creates a right-angled triangle aligned to the left using stars.


```
// Pyramid pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = 1; i <= n; i++) {

            // inner loop to handle columns
            for (j = 1; j <= i; j++) {
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }
```

```
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

13. Reverse Right Half Pyramid Pattern

This prints a right-aligned triangle but in reverse top to bottom.

```
// Java Program to print pattern
// Reverse Right Half Pyramid
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = n; i >= 1; i--) {

            // inner loop to handle columns
            for (j = 1; j <= i; j++) {
```

```
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

14. Left Half Pyramid Pattern

This pattern aligns a triangle to the right by adding spaces before stars.

```
// Java Program to print pattern
// Left Half Pyramid pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;
```

```
        // outer loop to handle rows
        for (i = n; i >= 1; i--) {

            // inner loop to print spaces.
            for (j = 1; j < i; j++) {
                System.out.print(" ");
            }

            // inner loop to print stars.
            for (j = 0; j <= n - i; j++) {
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

15. Reverse Left Half Pyramid Pattern

This prints a left pyramid in reverse, starting from full-width and decreasing.

```java
// Java Program to print pattern
// Reverse Left Half Pyramid
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = n; i > 0; i--) {

            // inner loop to print spaces.
            for (j = 0; j < n - i; j++) {
                System.out.print(" ");
            }

            // inner loop to print stars.
            for (j = 0; j < i; j++) {
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

16. Triangle Star Pattern

This prints an equilateral triangle where stars are spaced apart.

```java
// Java Program to print
// Triangular Pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = 0; i < n; i++) {

            // inner loop to print spaces.
            for (j = n - i; j > 1; j--) {
                System.out.print(" ");
            }

            // inner loop to print stars.
            for (j = 0; j <= i; j++) {
                System.out.print("* ");
            }

            // printing new line for each row
```

```
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

17. Reverse Number Triangle Pattern

A reverse right-aligned triangle with ascending numbers from i to n.

```
// Java Program to print pattern
// Reverse number triangle
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = 1; i <= n; i++) {
```

```
            // inner loop to print spaces.
            for (j = 1; j < i; j++) {
                System.out.print(" ");
            }

            // inner loop to print value of j.
            for (j = i; j <= n; j++) {
                System.out.print(j + " ");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
1 2 3 4 5 6
 2 3 4 5 6
  3 4 5 6
   4 5 6
    5 6
     6
```

18. Mirror Image Triangle Pattern

This creates a mirrored triangle with numbers, forming a diamond-like shape.

```
// Java Program to print pattern
// Mirror Image of a triangle
```

```
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // Printing the upper part
        for (i = 1; i <= n; i++) {

            // inner loop to print spaces.
            for (j = 1; j < i; j++) {
                System.out.print(" ");
            }
            // inner loop to print value of j.
            for (j = i; j <= n; j++) {
                System.out.print(j + " ");
            }

            // printing new line for each row
            System.out.println();
        }

        // Printing the lower part
        for (i = n - 1; i >= 1; i--) {

            // inner loop to print spaces.
            for (j = 1; j < i; j++) {
                System.out.print(" ");
            }

            // inner loop to print value of j.
            for (j = i; j <= n; j++) {
                System.out.print(j + " ");
            }

            // printing new line for each row
            System.out.println();
        }
```

```
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
1 2 3 4 5 6
 2 3 4 5 6
  3 4 5 6
   4 5 6
    5 6
     6
    5 6
   4 5 6
  3 4 5 6
 2 3 4 5 6
1 2 3 4 5 6
```

19. Hollow Triangle Pattern

This prints a hollow equilateral triangle using stars with only the boundary.

```
// Java Program to print
// Hollow triangle pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
```

```
        int i, j, k;

        // outer loop to handle rows
        for (i = 1; i <= n; i++) {

            // inner loop to print spaces.
            for (j = i; j < n; j++) {
                System.out.print(" ");
            }

            for (k = 1; k <= (2 * i - 1); k++) {

                // printing stars.
                if (k == 1 || i == n || k == (2 * i - 1)) {
                    System.out.print("*");
                }

                // printing spaces.
                else {
                    System.out.print(" ");
                }
            }

            System.out.println("");
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

```
     *
    * *
   *   *
```

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

20. Hollow Reverse Triangle Pattern

Prints an inverted hollow triangle with stars only at boundaries and top row.

```java
// Java Program to print pattern
// Reverse Hollow triangle
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j, k;

        // outer loop to handle rows
        for (i = n; i >= 1; i--) {

            // inner loop to print spaces.
            for (j = i; j < n; j++) {
                System.out.print(" ");
            }

            for (k = 1; k <= (2 * i - 1); k++) {

                // printing stars.
                if (k == 1 || i == n || k == (2 * i - 1)) {
                    System.out.print("*");
                }

                // printing spaces.
                else {
                    System.out.print(" ");
                }
            }
```

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

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

21. Hollow Diamond Pyramid

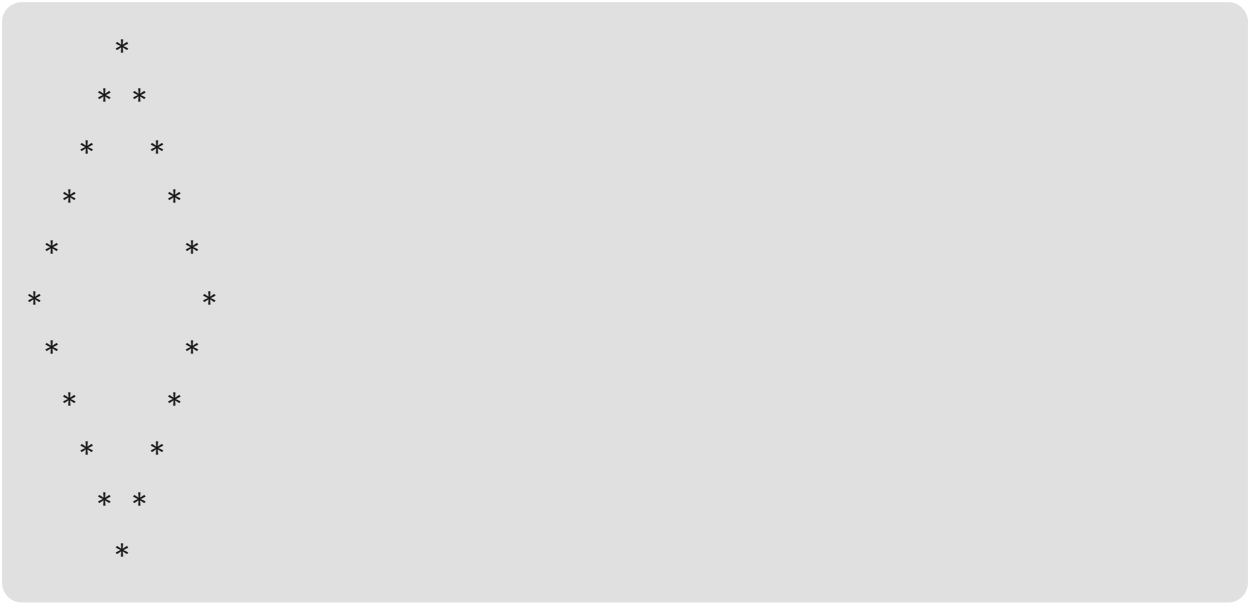
Prints a symmetrical hollow diamond made of stars, with spaces in the middle and stars only at the borders.

```
// Java Program to print Pattern
// Hollow Diamond Star
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle upper part
```

```
        for (i = 1; i <= n; i++) {

            // inner loop to print spaces
            for (j = 1; j <= n - i; j++) {
                System.out.print(" ");
            }

            // inner loop to print stars
            for (j = 1; j <= 2 * i - 1; j++) {
                if (j == 1 || j == 2*i-1)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }

        // outer loop to handle lower part
        for (i = n-1; i >= 1; i--) {

            // inner loop to print spaces
            for (j = 1; j <= n - i; j++) {
                System.out.print(" ");
            }

            // inner loop to print stars
            for (j = 1; j <= 2 * i - 1; j++) {
                if (j == 1 || j == 2*i-1)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
```

```
    }
}
```

Output



22. Hollow Hourglass Pattern

Displays a hollow hourglass using stars, where only boundary stars are visible in a symmetric hourglass shape.

```
// Java Program to print pattern
// Hollow Hourglass Pattern
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // Printing the upper part
        for (i = 1; i <= n; i++) {

            // inner loop to print spaces.
            for (j = 1; j < i; j++) {
```

```
                System.out.print(" ");
            }

            // inner loop to print value of j.
            for (j = i; j <= n; j++) {
                if(j==i||j==n||i==1)
                    System.out.print("* ");
                else
                    System.out.print("  ");
            }

            // printing new line for each row
            System.out.println();
        }

        // Printing the lower part
        for (i = n - 1; i >= 1; i--) {

            // inner loop to print spaces.
            for (j = 1; j < i; j++) {
                System.out.print(" ");
            }

            // inner loop to print value of j.
            for (j = i; j <= n; j++) {
                if(j==i||j==n||i==1)
                    System.out.print("* ");
                else
                    System.out.print("  ");
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
```

```
    }
}
```

Output

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

23. Pascal's Triangle

Generates Pascal’s Triangle with proper alignment, showing binomial coefficients for each row.

```
// Java Program to implement
// Pascal's Triangle
import java.util.*;

class Geeks {

    // Pascal function
    public static void printPascal(int n)
    {
        for (int i = 1; i <= n; i++) {
            for (int j = 0; j <= n - i; j++) {

                // for left spacing
                System.out.print(" ");
            }
```

```
            // used to represent x(i, k)
            int x = 1;
            for (int k = 1; k <= i; k++) {

                // The first value in a line is always 1
                System.out.print(x + " ");
                x = x * (i - k) / k;
            }
            System.out.println();
        }
    }

    // Driver code
    public static void main(String[] args)
    {
        int n = 4;
        printPascal(n);
    }
}
```

Output

```
   1
  1 1
 1 2 1
1 3 3 1
```

24. Right Pascal's Triangle

Forms a right-angled triangle pointing upward and downward, creating a right Pascal's triangle shape.

```
// Java Program to print
// Right Pascal's Triangle
import java.util.*;

// Java code for printing pattern
public class Geeks {
```

```
    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;
        int num = 1;

        // outer loop to handle upper part
        for (i = 1; i <= n; i++) {

            // inner loop to print stars
            for (j = 1; j <= i; j++) {
                System.out.print("* ");
            }

            System.out.println();
        }

        // outer loop to handle lower part
        for (i = n-1; i >= 1; i--) {

            // inner loop to print stars
            for (j = 1; j <= i; j++) {
                System.out.print("* ");
            }

            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 4;
        printPattern(n);
    }
}
```

Output

```
*
* *
```

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

25. K Pattern

Prints an inverted and then upright right half pyramid (like K shaped), forming a symmetrical star pattern.

```java
// Java Program to print pattern
// Reverse Right Half Pyramid
import java.util.*;

public class Geeks {

    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j;

        // outer loop to handle rows
        for (i = n; i >= 1; i--) {

            // inner loop to handle columns
            for (j = 1; j <= i; j++) {
                System.out.print("*");
            }

            // printing new line for each row
            System.out.println();
        }

        // outer loop to handle rows
        for (i = 2; i <= n; i++) {

            // inner loop to handle columns
            for (j = 1; j <= i; j++) {
                System.out.print("*");
```

```
            }

            // printing new line for each row
            System.out.println();
        }
    }

    // Driver Function
    public static void main(String args[])
    {
        int n = 6;
        printPattern(n);
    }
}
```

Output

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

Conclusion

Java pattern programs are a great way to learn and practice coding skills. They help you understand loops, nested loops, and how to think logically to solve problems. Whether you are a beginner or an experienced programmer, practicing pattern programs can improve your Java skills. So, keep coding, experimenting with different patterns, and enjoy the learning process

Next Article

Printing Pyramid Patterns in Python

Similar Reads

1. Programs for printing pyramid patterns in Java
2. Java Program to Print a New Line in String
3. Java Program to Print Left Triangle Star Pattern
4. Java Program to Print Downward Triangle Star Pattern
5. Java Program to Print Mirror Lower Star Triangle Pattern
6. Java Program to Print Diamond Shape Star Pattern
7. Java Program to Print Square Star Pattern
8. Program to print numbers with diamond pattern
9. Java Program to Print Pyramid Star Pattern
10. Java Program to Print Mirror Upper Star Triangle Pattern

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

NumPy
NLP
Deep Learning

NextJS
NodeJs
Bootstrap
Tailwind CSS

Python Tutorial

Python Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

Databases

SQL
MYSQL
PostgreSQL
PL/SQL
MongoDB

Preparation Corner

Company-Wise Recruitment Process
Aptitude Preparation
Puzzles
Company-Wise Preparation

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets

Courses

IBM Certification Courses
DSA and Placements
Web Development
Data Science

Programming Languages

C Programming with Data Structures
C++ Programming Course
Java Programming Course
Python Full Course

Programming Languages

DevOps & Cloud

Clouds/Devops

DevOps Engineering

AWS Solutions Architect Certification

Salesforce Certified Administrator Course

GATE 2026

GATE CS Rank Booster

GATE DA Rank Booster

GATE CS & IT Course - 2026

GATE DA Course 2026

GATE Rank Predictor