



Patterns

Created	@January 18, 2026 5:44 PM
Class	@January 18, 2026 5:45 PM

Patterns:

Q. solid rectangle using *

```
int m= user no.;  
int n=user no.;  
//Outer loop  
}  
for(int i=1; i≤n; i++){  
    //inner loop  
    for(int j=1; j≤m; j++){  
        System.out.print("*");  
    }  
    System.out.println("*");  
}
```

first we have taken the n and m from the use i.e. the stoping value of loop then the for loop is use and here i and j is used for the starting value for making it into row and column i is for row and j is for column inner for loop used the only print for not having the spaced line and outer line can be used the println outer loop is used for rows and inner loop is used for columns

Q. hollow rectangle

only border have asterisks

```

for(int i=1; i≤n; i++){
    for(int j=1; j≤m; j++){
        if(i==1 || j==1 || i==n || j==m){
            System.out.print("*");
        }else{
            System.out.print(" ");
        }
    }
    System.out.println();
}

```

here for creating the hollow rectangle the matrix is used for creating the if-else condition is used and the if condition is given with or operator

Q. half pyramid

```

for(int i=1; i≤n; i++){
    for(int j=1; j≤i; j++){
        System.out.print("*");
    }
    System.out.println();
}

```

Q. inverted half pyramid

```

for (int i=n; i≥1; i- -){
    for(int j=1; j≤i; j++){
        System.out.print("*");
    }
    System.out.println();
}

```

here we inverted the pyramid just by doing the changes in condition i.e. $i=n$ means i has the value of n , $i \geq 1$ means i is less than equal to 1 and i is decrementation.

Q. half pyramid but from left side

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

the outer loop is for the rows first inner loop is for the printing the space and 2nd inner loop is for printing the *.

Q. half pyramid with numbers

```
for(int i=1; i<=n; i++){
    for(int j=1; j<=i; j++){
        System.out.print(j+" ");
    }
    System.out.println();
}
```

here we just pretend j in inner loop for printing the next number in order

Q. Floyd's triangle

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

```

```

        System.out.print(number+" ");
        number++;
    }

    System.out.println();
}

```

In this triangle the numbers goes on continuing like (1,2,3,4) through each row. here we have taken the new variable 'number' for the printing the next numbers in sequence

Q. Binary Triangle

```

for(int i=1; i≤n; i++){
    for(int j=1; j≤i; j++){
        if(sum%2==0){
            System.out.print("1 ");
        }else{
            System.out.print("0 ");
        }
        System.out.println();
    }
}

```

- Advanced Patterns

Q. Butterfly patterns

for(int i=1 to n)

1st part stars i.e. rows

for(int j=1 to i) }

spaces= 2*(n-1)

for(int j=1 to i) }

2nd part stars

for(int j=1 to i) }

2nd half (n to 1)

i=n=4

code:

for(int i=1; i≤n; i++){

```

for(int j=1; j≤i; j++){
    System.out.print("*");
}

int spaces=2*(n-i);
for(int j=1; j≤spaces; j++){
    System.out.print(" ");
}

for(int j=1; j≤i; j++){
    System.out.print("*");
}

System.out.println();
}

for(int i=n; i≥1; i- -){
    for(int j=1; j≤i; j++){
        System.out.print("*");
    }

    int spaces=2*(n-i);
    for(int j=1; j≤spaces; j++){
        System.out.print(" ");
    }

    for(int j=1; j≤i; j++){
        System.out.print("*");
    }

    System.out.println();
}

```

Q. solid rhombus

n= no. of rows
 m=no. of stars
 just focusing on the spaces
 outer loop
 for (int i=1; i≤n; i++){
 for(int j=1; j≤n-i; j++){
 System.out.print(" ");
 }
 for(int j=1; j≤m; j++){
 System.out.print("*")
 }
 System.out.print();
 }

Q. for hollow solid rhombus

```

for(int i=1; i≤n; i++){
    for(int j=1; j≤n-i; j++){
        System.out.print(" ");
    }
    for(int j=1; j≤n; j++){
        if(i==1 || i==n || j==1 || j==n){
            System.out.print("*");
        }else{
            System.out.print(" ");
        }
    }
    System.out.println();
}
  
```

Q. number pyramid

pyramid of numbers

```
1  
2 2  
3 3 3  
4 4 4 4
```

- | | |
|-------------------------------|-----------------------|
| i=1 then no. 1 printed 1 time | 1st row 3 spaces no.1 |
| i=2 then no.2 printed 2 time | 2nd row 2 spaces no.2 |
| i=3 then no.3 printed 3 time | 3rd row 1 space no.3 |
| i=4 then no.4 printed 4 time | 4th row 0 space no.4 |

```
for(int i=1; i≤n; i++){  
    for(int j=1; j≤n-i; j++){  
        System.out.print(" ");  
    }  
    for(int j=1; j≤i; j++){  
        System.out.print(i+" ");  
    }  
    System.out.println();  
}
```

Q. Palindrome pattern

means the word from front and backwards is the same word e.g. BOB, level, etc.

print the palindrome in no. like 212, 32123,etc.

```
for(int i=1; i≤n; i++){  
    for(int j=1; j≤n-i; j++){  
        System.out.print(" ");  
    }  
    for(int j=i; j≥1; j--){  
        System.out.print(j);  
    }
```

```
for(int j=2; j≤i; j++){\n    System.out.print(j);\n}\nSystem.out.println();\n}
```

Q. diamond pattern

here stars=2*i-1

if i=1 then stars=2*1-1=1

if i=2 then stars=2*2-1=3

and so on

```
for(int=i; i≤n; i++ {\n    for(int j=1 j≤n-i; j++{\n        System.out.print(" ");\n    }\n    for(int j=1; j≤2*i-1; j++{\n        System.out.print("*");\n    }\n    System.out.println();\n}\nfor(int=n; i≥i; i- - {\n    for(int j=1 j≤n-i; j++{\n        }\n        System.out.print(" ");\n    for(int j=1; j≤2*i-1; j++{\n        System.out.print("*");\n    }\n    System.out.print();\n}
```

}