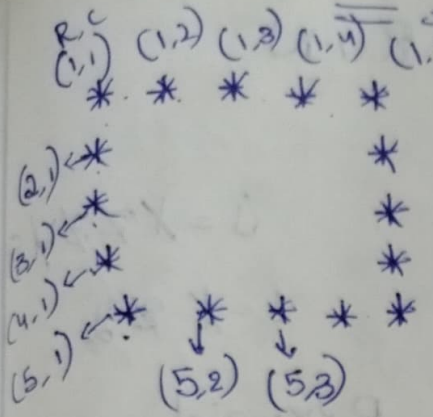
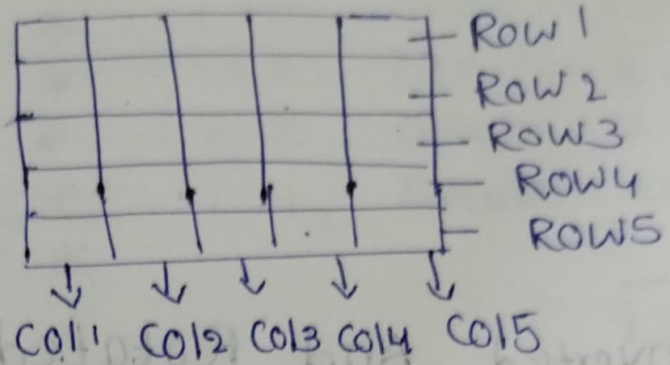
Hollow Rectangle Pattern





Size = rows X cols

5 X 5

boundary

rows → 1, 4

col → 1, 5

logic

1) Total lines (total rows)

outer loop (1 to 5)

(n)

↳ tot rows

2) boundary condition

(row == 1 || col == 1 || row == 4 || col == 5)

inner

```
//outer loop
for (int i = 1; i <= totrows; i++) {
  //inner - col
  for (int j = 1; j <= totcol; j++) {
//cell - (i, j)
  if (i == 1 || i == totrows || j == 1 || j == totcol) {
        // boundary conditions
    System.out.Print("*");
    }
  else { System.out.Print(" "); } }
```

System.out.println();
}

Dry RUN

i = 1 j = 1 2 3 4 5
i = 2 j = 1 2

Inverted And Rotated half Pyramid

		Spaces	Stars	
r=1	- - - *	3 → 4-i	1	i = 1
r=2	- - * *	2 → 4-i	2	i = 2
r=3	- * * *	1 → 4-i	3	i = 3
r=4	* * * *	0 → 4-i	4	i = 4

spaces = 4 - i
= n - i

0-1 Triangle

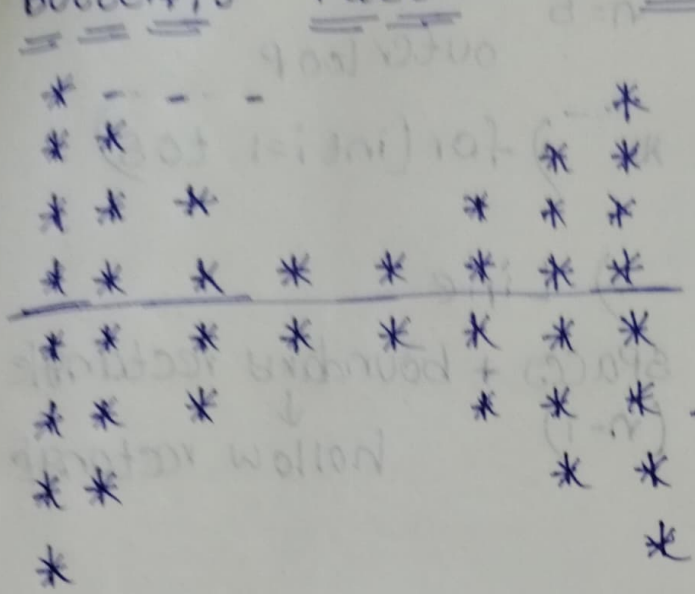
(1,1) → 1+1 = 2 → 1

odd 1 → i=1 j=1 to 1
(2,1) 0 1 → i=2 j=1 to 2
(3,1) 1 0 1 → i=3 j=1 to 3
(4,1) 0 1 0 1
(5,1) 1 0 1 0 1 → (5,5)

i → outer loop → Row
j → inner loop → col

(i+j) → even → "1"
(i+j) → odd → "0"

Butterfly Pattern n=4



outer loop 1st half

1) for i = 1 to i <= n ; i → rows

2) Line

stars + spaces + stars
i i

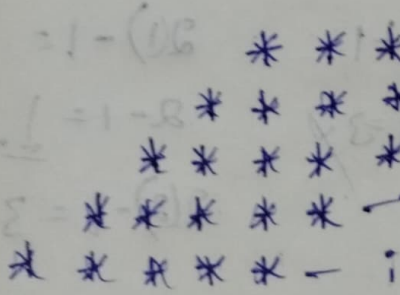
spaces = 2 x (n-i)

2nd half → inverted

for (int i = n; i >= 1; i--)

inner loop → same

Solid Rhombus



outer loop
↳ n=5

for (int i=1; i<=n; i++)

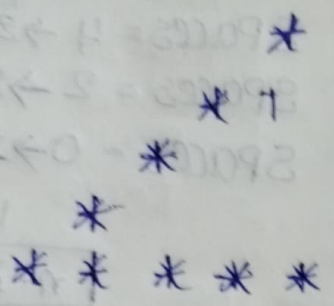
stars = (j=1 to n)

spaces = n - i

n-i
5-4

Hollow Rhombus

n = 5



outer loop

1) for (int i=1 to 5)

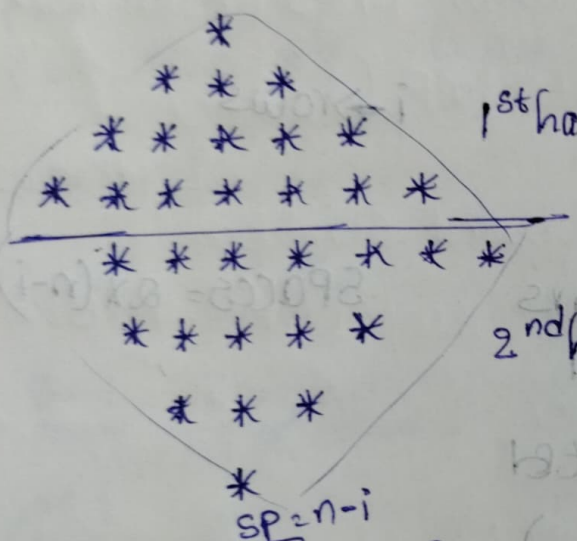
2) Line

spaces + boundary rectangle

(n-i) ↓ hollow rectangle

Diamond Pattern

n = 4



1st half

1) outer loop

for (int i=1; i<=n; i++)

2) What to Print

i=1	SP=3	St=1 ✓
i=2	SP=2	St=3 ✓
i=3	SP=1	St=5
i=4	SP=0	St=7

odd

2x+1 2x-1

2(1)+1 = 2+1 = 3 ✗

2(1)-1 = 2-1 = 1 ✓

2(2)-1 = 3 ✓