int[] name

String[] name

new string [3];

name[0] = "AK";

name[1] = "JK";

name[2] = "PK";

sysout (name[0])

(1)

```
import java.util.*;
Public class conditions {
Public static void main (String[] args) {

Scanner sc = new scanner (system.in);
int age = sc.newInt();

If (age > 18) {
sysout ctrl+space ("Adult");
}
else {
sysout ctrl+space ("Not an Adult");
}
}
}
```

(2)

```
import java.util.*;
public class condns {
Public static void main (String[] args){
Scanner sc = new scanner (system.in)
int x = sc.nextInt();
if (x%==2) {
sysout ("even")
}
else {
{("Odd");
}
}
}
```

print enter your no:

```
import java.util.*;
public class conditions {
public static void main (String[] args) {
Sc.Scanner = new scanner (system.in)
int a = sc.nextInt();
int b = sc.nextInt();
If (a==b) {
Sysout ("Equal");
} else {
If (a>b) {
Sysout ("a is greater");
}
else {
Sysout ("b is greater");
}
}
}
}
```

Quiz

If 3 is an odd number, print "Bazinga".

A.
```
if (3%2==0) {
} Sysout ctrl+space ("Bazinga");
```

B)
```
If (3%2==0)
Sysout ctrl+space ("Bazinga");
```

C)
```
If (3%2==0);
Sysout ctrl+space ("Bazinga");
```

```
switch (variable) {        → number
                           → char 'a', 'b', 'c'
case1 :
        stt1;
        break;
case2 :
        stt2;
        break;
case 3 :
        stt 3;
        break;
default :
        stt ;
}
```

Lec 4. Loops

```
for (intialisation; condition; updation) {
    statement
}
```

- intialisation: int counter = 0 variable
- condition: counter = counter + 1 ; counter <= 100
- updation: always while updation counte = reunter + 1

```
public class Iterative loops {
    public static void main (String[] args) {
        for ( counter = 0 , counter < 100 , counter = counter + 1 ){
            System.out.println (" I'm Akanksha");
        }
    }
}
```

Array.

```
int [] note = new int note [4];
note[0] = 30;
note[1] = 31;
note[2] = 56;
note[3] = 49;
```

Q5 Print the Numbers from 0 to 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

i = 0 initial
i = 1 increment
i = 2
...
i = 10
i = 11 termination

DRY RUN

```
for (int i = 0; i < 11; i++) {
    print (i);
}
```

```
while (condition) {
    // do something
}
```

```
Public class WhileLoop {
    Public static void main (String[] args) {
        int i = 0;
cond —  while (i < 11) {
        i = i + 1;
        Sysout ctrl + space ("I'm Programmer");
    }
}
```

1.

Kuch karo until while gets satisfied

```
do while {
```

First print then condn checking

aaff in dowhile & while

condn checking main printing.

If Kaam ko atleast 1 bar karna hai toh do while use karo.

```
int i=0;
do {
    print (i);
    i=i+1;
} while (i<11);
    i = i + 1;
```

DRY RUN

i = 0
i = 1
2
3
...
10

```
Public class DoWhile {
Public static void main (String[] args) {
    Do {
        int a = 0;
        Do {
            Sysout ctrl + space (a); a = a+1;
        } while (a<11);
```

Sum of 1st n natural numbers

```
int sum = 0;
for (i=1; i<=n; i++){
    sum = sum+i;
} print (sum);
}
```

```
import java.util.*;
public class for loop {
public static void main(String [] args){
  scanner sc = new scanner (system.in);
  int n = sc.nextInt();
  ~~int~~ sum = 0;
  for (int i = 1; i<=n; i++){
    sum = sum+i;
    s.ysout ctrl+space (sum);
}
```

Print the table of a no. I/p by user

```
input n    int n;
for (int i = 1; i<11; i++){
  ~~i = i * n;~~
  sysOut ctrl+space (i*i);
}
```

```
import java.util.*;
public class Conditional statement {
public static void main (String [] args){
 scanner.sc = new Scanner (System.in);
 int n = sc.nextInt();
 sysout ctrl+space("Enter the number:");
 for (int i=1; i<11; i++){
  sysout ctrl+space (i*n)
 }
}
}
```

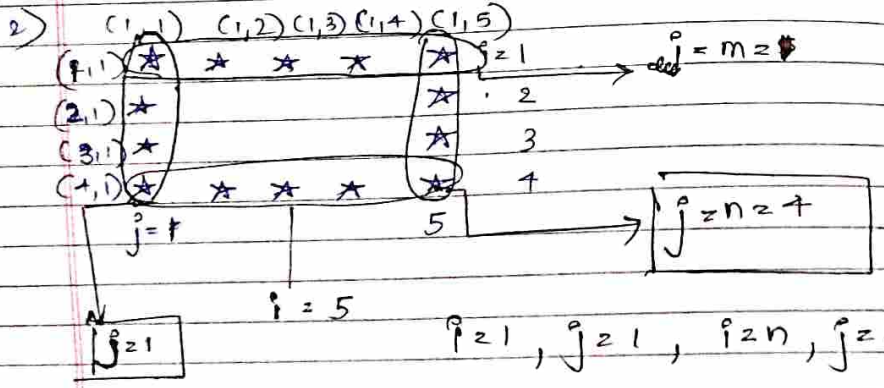
pattern questions

```
public class Loops {
public static void main
    (String[] args){
    int n = 4;
    int m = 5;
    for (int i = 1 ; i <= n; i++ ){
        for (int j = 1; j <= m; j++) {
            sysout ctrl+space ("*");
        }
        system.out.println();
    }
}
```

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

rows - Outer loops
columns - Inner loops

2)



i=1, i<n → outer loop = rows
j=1, j<m → inner loop = column

```
public class Pattern {
public static void main (String[] args){
    n = 4; // rows
    m = 5; //——1(column)
for (int i=1; i<=n ; i++) {
for (int j=1 ; j<=m; j++) {
    if ( i==1 || j==1 || j==n || j==m) {
        system.out.print("*");
    } else {
        system.out.print(" ");
    }
}
system.out.println();
}
}
```

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

Half pyramid.

```
n = 4
——
——
for (int i = 1; i<=n ; i++) {
    for( int j=1 ; j<=i ; j++) {
        System.out.print("*");
    }
    System.out.println();
}
}
```

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

```
for(int i = 1; i<=4; i++){
    for(int j=1; j<=i; j++)
        System.out.print("*")
    }
    System.out.println();
}
```

DRY RUN

i=1 → j=1

j++ → 2

i=2 → j=2

i=3 → j=1,2,3

i=4 → j=1,2,3,4

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

```
for (int i=n; i>=1; i--){
    for(int j=1; j<=i; j++)
        print (*)
    }
    println();
}
```

```
public class Pattern {
public static void main (String [] args){
    int n=4;
    for (int i=n; i>=1; i--) {
        for(int j=1; j<=i; j++) {
            System.out.
```

5)

```
n-1   n=1   *      <-- 3 space + 1 star --> 4
n-2   n=2   * *    <-- 2 space + 2 star --> 4
n-3   n=3   * * *  <-- 1 space + 3 star --> 4
n=4   n=4   * * * * <-- 0 space + 4 star --> 4
```

n = no. of rows

n=4;

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

o/p

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

Logic fo

① n-fs 1
1 * will be printed

② ~~i++~~ i will get increment

③ i=1; i<=n
will not get satisfied

④ comes out from inner for loop

⑤ prints new line

⑥ Again comes in outer for loop

⑦ now n=2 in inner for loop

⑧ prints → 2 *

⑨ again after increment n gets 3 dissatify the condn

⑩ for comes out from inner For loop

⑪ prints new line

*	3 space + 1 star	
* *	2 space + 2	
* * *	1 + 3	
* * * *	0 + 4	

for (i=1 to n)
→ space
for (j=1 to i)
*

n-1 ... i=1

n-1 * ... 2
* * *
* * * *

n=1
"*"

for (i=1; j<=n; i++)
for (i=1 to n)
SPACE

for (j=1; j==i; j++)
print ("*")

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

① first print rows
② 2nd print ~~columns~~ spaces
③ 3rd print *

Pattern

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

```
int n = 5;
//outer loop
for(int i=1; i<=n; i++){
    for(int j=1; j<=i; j++){
sys.print("j") or
    Sysout ctrl+space ("j"+" ");
}
    Sysout ctrl+space ();
}
```

Inverted half pyramid with nos

```
1 2 3 4 5   <- r=1  1 to 5
1 2 3 4     r=2    1 to 4
1 2 3       r=3    1 to 3
1 2                1 to 2
1                  1 to 1
```

row ↑ column ↓ by -1

Print the pattern

```
1 2 3 4 5   r=1  i=1  1 to 5
1 2 3 4     r=2  i=2  1 to 4
1 2 3       r=3  i=3  1 to 3
1 2         r=4  i=4  1 to 2
1           r=5  i=5  1 to 1
```

```
for (i=1 to n)
    for(j=1 to n-i+1)

//outer loop
n=5
for(int i=1; i<=n; i++){
  for(int j<=1; j<=n-i+1; j++){
  sysout print("*");
}
  Sysout println();
}
}
```

Print pattern

```
1                  i=1  1      Floyd's Triangle
2 3                i=2  2 val  (2,3)
4 5 6              i=3  3 val  (4,5,6)
7 8 9 10           i=4  4 val  (7,8,9,10)
11 12 13 14 15     i=5  5 val  (11,12,13,14,15)
```

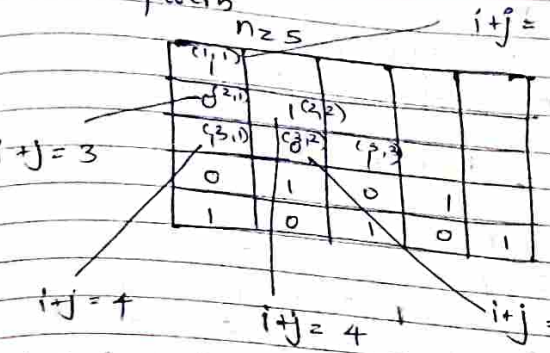
```
Public class Patten {
Public static void main (String [] args) {
int n = 5;
int number = 1;

//outer loop
for(int i=1; i<=5; i++) {
    for (int j=1; j<=i; j++) {
        sysout print (j);
        sysout  ctrl+space  (number++)
        number++;  number = number + 1;
    }
    System.out.println();
}
}
}
```

forrows → kitne number print hone hai

number (+1)

1)

Print the pattern



Even — 1
Odd — 0

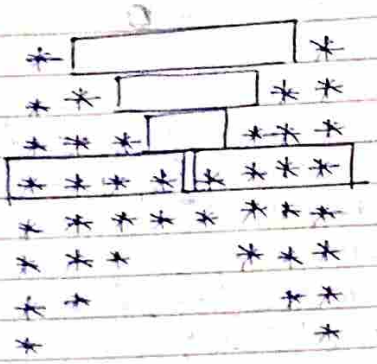
```
int n = 5;
for (int i=1; i<=n; i++){
  for(int j=1; j<=i; j++){
    int sum = i+j
    If (sum %2 == 0) {
      print ("1");
    }
    Else {
      print ("0");
    }
  }
  System.out.println();
}
}
}
```

DRY RUN

```
for ( i  1 to n )
  for ( int j=1 to i)
        i + j = even   "1"
        i + j = odd    "0"
```

i = 1 , j = 1 → 1
i = 2 , j = 1, 2 → 0, 1
i = 3 , j = 1, 2, 3 → 1, 0, 1
i = 4 , j = 1, 2, 3, 4 → 0, 1, 0, 1
i = 5 , j = 1, 2, 3, 4, 5 → 1, 0, 1, 0, 1

10)



row no.			
1	1 star	6 spaces	1 star
2	2 star	4 spaces	2 stars
3	3 star	2 spaces	3 stars.
4	4 stars	0 spaces	4 stars

① step

Variable declaration is limited upto its scope

To print star

1	1 star	6 spaces	1 star
2	2 star	4 spaces	2 star
3	3 star	2 spaces	3 star
4	4 stars	0 spaces	4 star

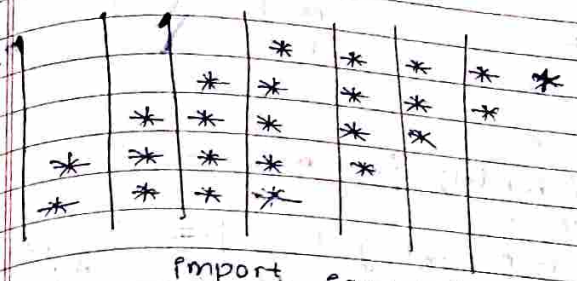
i = 1
i = 2
i = 3
i = 4

2 * 3 → n-i (4-1)
2 * 2 → n-i (4-2)
2 * 1 → n-i (4-3)
2 * 0 → n-i (4-4)

spaces = (2 * (n-i))

```
for (int i  1 to n)        ---- starting rows
// 1st part stars
for ( int j=1 to i)        ---- row no.
  spaces = 2*(n-1)
  for (int j=1 to spaces)  ---- spaces
// 2nd part stars
for(int j=1 to i)          ---- 2nd part of stars
```

```
import java.util.*;

public class patterns {
    public static void main (String args [] ){
        int n = 5;
        // first half - upper half
        for (int i = 1 ; i <= n; i++ ) {
            // 1st part
            for ( int j = 1 ; j <= i ; j++ ) {
                System.out.print ("* ");
            }
            // spaces
            int spaces = 2 * (n-i);
            for (int j=1 ; j <= spaces ; j++)
                system.out.print (" ");
            // 2nd part
            for (int j=1 ; j<=i ; j++ ){
                System.out.print ("* ");
            }
            System.out.println ();
        }

        for (int i=n; i >= 1 ;i--){
            // 1st part
            for (int j=1 ; j<=i ; j++){
                system.out.print ("* ");
            }
            // spaces
            int spaces = 2 * (n-i);
            for (int j=1 ; j <= spaces ; j++)
                System.out.print (" ");
            // 2nd
            for (int j=1 ; j <= i ; j++){
                System.out.print ("* ");
            }
            System.out.println ();
        }
    }
}
```



```
import java.util.*;
public class Pattern4 {
    public static void main (String args [] ){
        int n = 5;
        for (int i=1 ; i <= n; i++ ) {
            // spaces
            for (int j = 1; j<= n-i; j++ ) {
                System.out.print (" ");
            }
            // stars
            for ( int j = 1; j<= 5 ; j++ ) {
                System.out.print ("* ");
            }
            System.out.println();
        }
    }
}
```

Print the pattern

n = 5

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

Spaces

n-1 — 4 — i=1
n-2 — 3 — i=2
n-3 — 2 — i=3
n-4 — 1 — i=4
n-5 — 0 — i=5

```
int 5
for (int i=1; i<=n ; i++){
  // spaces
  for (int j=1 ; j<=n-i ; j++){
    System.out.print(" ");
  }
  // numbers
  for (int j=1 ; j<=i ; i++){
    System.out.print( i+" ");
  }
  sysoutprintln ( );
}
}
```


Print the pattern

Palindromic Pattern

```
    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
```

i=1 space - 4
i=2 3
i=3 2
i=4 1
i=5 0

1 to 1 spaces = (n-i)
2 to 1 for (j=1 to 1)
3 to 1 for (j=2 to 1)
4 to 1 (j=3 to 1)
5 to 1

```
import java.util.*;
public class Patterns {
  public static void main (String args[]) {
    int n = 5;
    for (int i=1; i<=n ; i++) {
      // spaces
      for (int j=1 ; j<=n-i ; j++) {
        System.out.print (" ");
      }
      // 1st half numbers
      for (int j=1 ; j>=1; j--) {
        System.out.print (j);
      }
      for (int j=2 ; j<=i ; j++) {
        System.out.print (j);
      }
      System.out.println ( );
    }
  }
}
```

14)

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

n = 4

i = 1	spaces = 3	1
i = 2	spaces = 2	3
i = 3	spaces = 1	5
		7

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

stars | 2 * i - 1

i = 1 star = 1
i = 2 stars = 2×2 - 1 = 3
i = 3 stars = 2×3 - 1 = 5

```
for (int i = 1 ; i <= n ; i++) {
    for (int j ; j <= 2*i-1 ; j++)
    for (int j ; j <= n-i ; j++)
        print (" ");
    }
    for (int j = 1; j <= 2*i-1 ; j++)
        print ("*");
    }
    Sysout println ();
}
for (int i = n ; i >= 1 ; i--) {
    for (int j ; = n-i ; j++)
        print (" ");
}
```

Patterns Trick

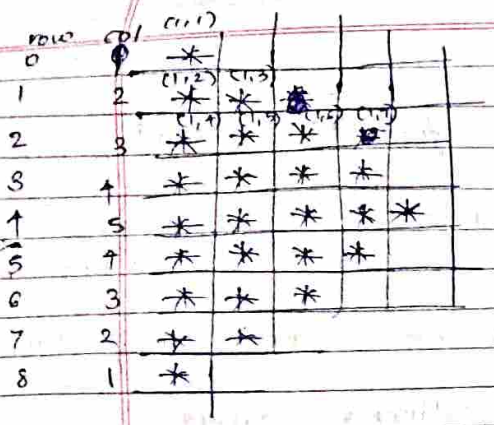
① Run outer for loop for lines that the pattern have
no. of lines = no. of rows = no. of times outer loop will run

② Identify for every row no., how many column are there
Types of elements in columns present.

~~for ex~~

③ what do you need to print.

n = 5



(2*n - 1)

if row no. > N :

if

n

*

(2*n-1)

2 x 5 - 5 - 1

10 - 5 = 4

for (int row = 1 ; row < 2 * n, row++) {

~~for (int~~ int totalColsInRow = row > n ? 2*n - row : row ;

for (int col = 0; col < totalColsInRow; col++) {

Sysout ("*");

}

}

~~package~~ .

public class main {

public static void main (String [] args) {

pattern17 (n: 5);

}

static void pattern30 (int n) {

for (int row = 1, row <= n; row++) {

for (int space = 0, space < n - row; space++) {

System.out.print (" ");

}

for (int col = row, col >= 1; col--) {

System.out.print (col + " ");

}

for (int col = 2, col <= row; col++) {

System.out.print (col + " ");

}

System.out.println();

}

+ + + + +