10/11/22 Loops And pattern programming

↳ Hyder Abbas.

① Difference between System.out.print() and println()

① System.out.Print() :- only prints the Content without switching to the next line after Executing this statement.

② System.out.Println() :- Prints the Content and switch to the next line after Execution of the statement.

② Pattern Programming :-

(i) write a program to print 5 stars ?

```
*
*
*
*
*
```

```
S.o.p("*");
S.o.p("*");
S.o.p("*");
S.o.p("*");
S.o.p("*");
```

Olp :-

```
*
*
*
*
*
```

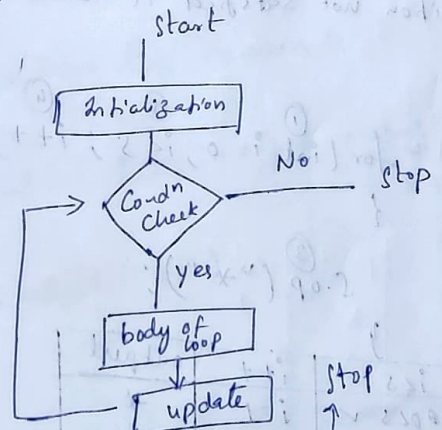
→ write program once and repeat it for multiple times.

③ Loops :- loop is used to iterate a part of the program several times. if the number of iteration is fixed.

→ Java provides three ways of Executing the loops

(i) for loop
(ii) While loop
(iii) do-while
(iv) For-Each (Enhanced for loop).
↳ discussed in Array Concept.

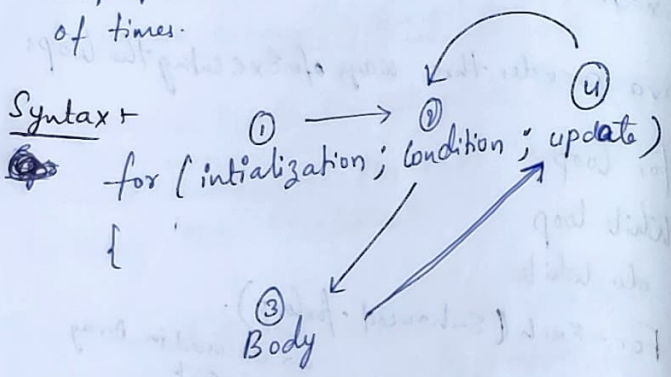
(i) for-loop :-



① Initialization → ② Condition checked ③ Body of loop ④ update

(i) **for loop** :- a for loop is used to repeat a specific block of code a known number of times.

Syntax :-



→ if condition not satisfied it comes out of loop.

Example ① :- for (int i = 0; i < 5; i++)

{

s.o.p ("*");

}

	i<5	i++	output
i = 0	0<5 ✓	i=1	*
i = 1	1<5 ✓	i=2	*
i = 2	2<5 ✓	i=3	*
i = 3	3<5 ✓	i=4	*
i = 4	4<5 ✓	i=5	*
i = 5	5<5 ✗	→	stops

i<5

0,1,2,3,4

0<5 → *
1<5 → *
2<5 → *
3<5 → *
4<5 → *

i<=4

0,1,2,3,4

0<=4 → *
1<=4 → *
2<=4 → *
3<=4 → *
4<=4 → *

Alter :-

```
int n = 5
for (int i=0; i<n; i++)
{
s.o.p ("*");
}
```

⇒ Print in same line *****

```
int n=5
for (int i=0; i<n; i++)
{
  s.o.Print ("*");
}
```

output: * * * * *

→ body of the loop executes if condition satisfies.

→ **Note** :- ① Intialization and update is optional for while and do-while.

(ii) While - loop :-

↳ As long As

→ While loop is a Control flow statement that allows code to be Executed repeatedly based on given Boolean Condition.

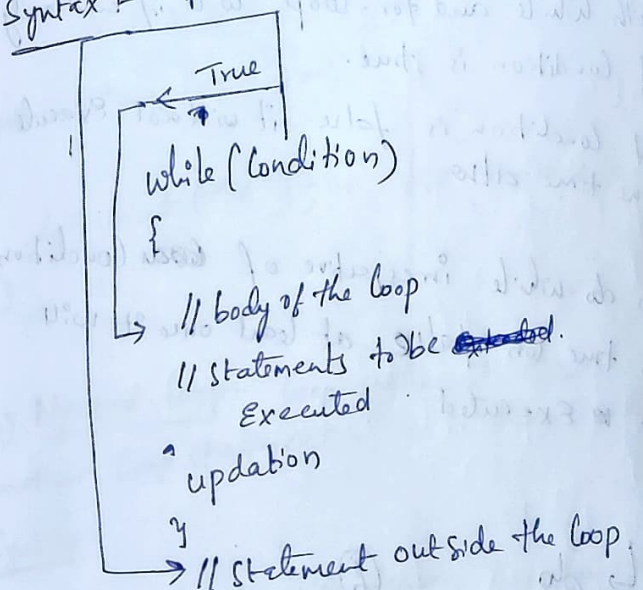
Eg:-
```
int i=0   ① → Initialization
int n=5
while(i<n) → ② Condition check
{
  S.o.print("*"); ③ → body
  i++;  → ④ updation
}
```

→ as long as i<n, do this task.

i	i<n	Print	i++
0	0<5	*	1
1	1<5	*	2
2	2<5	*	3
3	2<5	*	4
4	4<5	*	5
5	5<5 X		→ stops

output
*****

Syntax :-



→ initialization and updation is optional for while loop.

(iii) do while-loop :- is an Exit Control loop.

→ do-while loop check for the condition after Executing the statement of the loop body.

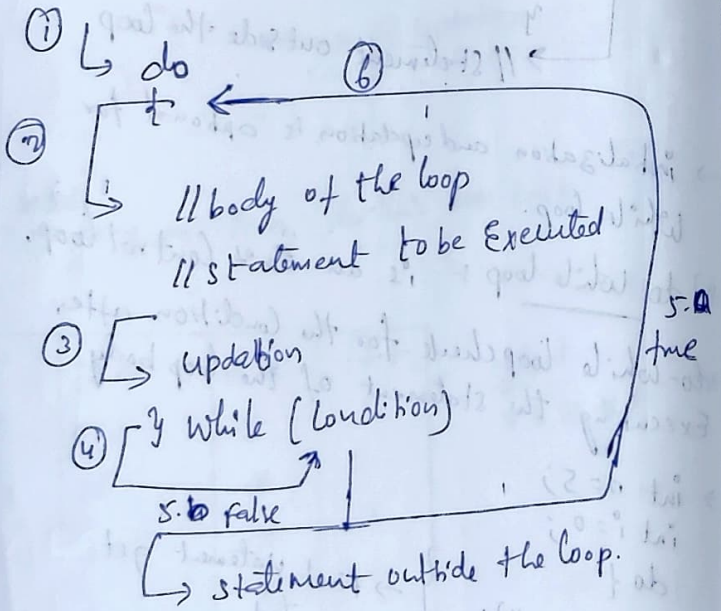
```
→ int n=5;
  int i=0;
  do {
  S.o.p("*");   } → first statement get Executed
    i++;
  } while(i<n); → Then Condition is checked.
```

Note:-

→ both while and for-loop will if and only if Condition is true.

→ if Condition is false it will not Execute one time also.

→ In do-while irrespective of Condition is true or false at least once it will be Executed.



① write a program to print

* * * * *

```
for(int i=0; i<5; i++)
{
   s.o.Print("*");
}
```

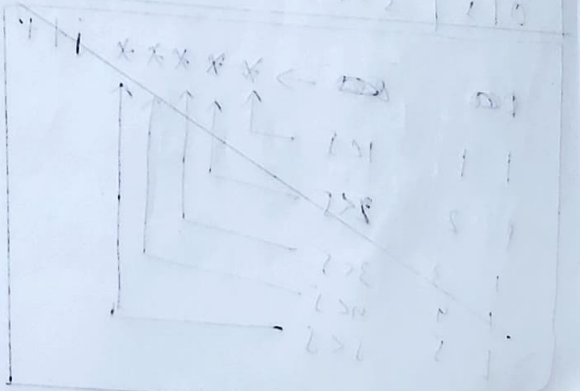
o/p:-

* * * * *

(iv) Nested Loop:- Loop statement inside another loop statement. it is also called

Syntax:-

```
for(initialization; Condition; increment){
   for(initialization; Condition; increment){
      // statement of inside loop.
   }
   // statement of outer loop.
}
```



Example ①

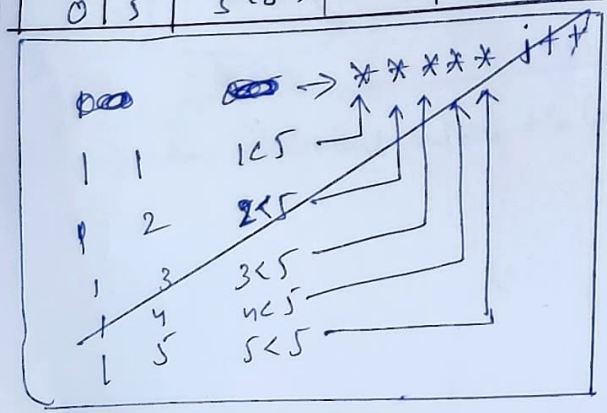
```
for(int i=0; i<5; i++)
{
for(int j=0; j<5; j++)
{
  S.o.Pfint(" * ");
}
}→ S.o.Println( );
```

for 1st Row

i	j	j<5	Print	j++
0	0	0<5 →	*****	1
0	1	1<5 →		2
0	2	2<5 →		3
0	3	3<5		4
0	4	4<5		5
0	5	5<5 X		

1st row.

→ stops.



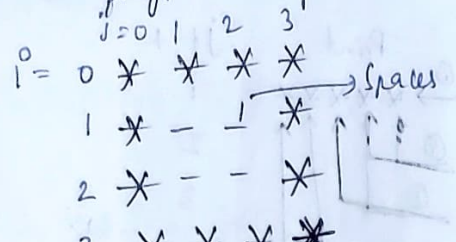
for second row.

i	j	j<5	Print	j++
1	0	0<5	① → *****	1
1	1	1<5	②	2
1	2	2<5	③	3
1	3	3<5	④	4
1	4	4<5	⑤	5
1	5	5<5 X	→ stops	

3rd rows.

2	0	0<5	→ *****	1
2	1	1<5		2
2	2	2<5		3
2	3	3<5		4
2	4	4<5		5
2	5	5<5 X	→ stops.	

① Write a program to print




```
int n=4;
for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        if(i==0 || i==n-1 || j==0 || j==n-1)
        {
            System.out.Print("*");
        }
        Else {
            System.out.Print("-");
        }
    }
    System.out.Println(" ");
}
```

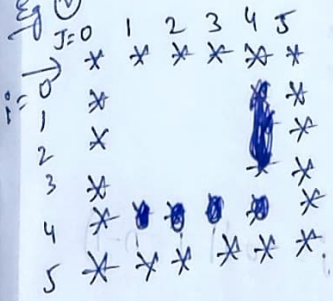
(n-1) (n-1) (Space)

→

```
int n=4;
if(int j=0 || j==n-1 || j==0 || j==n-1)
```

(on 3) (on 3)

Eg ⑤



i==0 || i==0 || j==0 || j==5

Eg② WAP to print

```
   j= 0 1 2 3 4 5
i=0  * * * * * *
1    *           *
2    *           *
3    *           *
4    *           *
5    * * * * * *
```

```
int n=6
if (i==0 || i==n-1 || j==0 || j==n-1)
{
   S.O.P("* ");
}
else{
   S.O.P("  ");
}
```

Eg③ WAP to print

```
   j= 0 1 2 3 4
i=0  *         *
1    *       * *
2    * * * * *  *
3    *         *
4    *
```

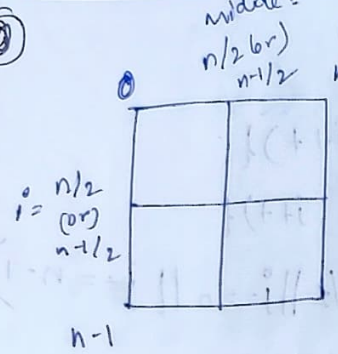
Code:

```
int n=6, i
for(int i=0; i<n; i++){
 for (int j=0; j<n; j++){
    if (i==(n-1)/2 || j==0 || j==n-1)
    {
       S.O.P("* ");
    }
    else{
       S.O.P("  ");
    }
 }
}
```

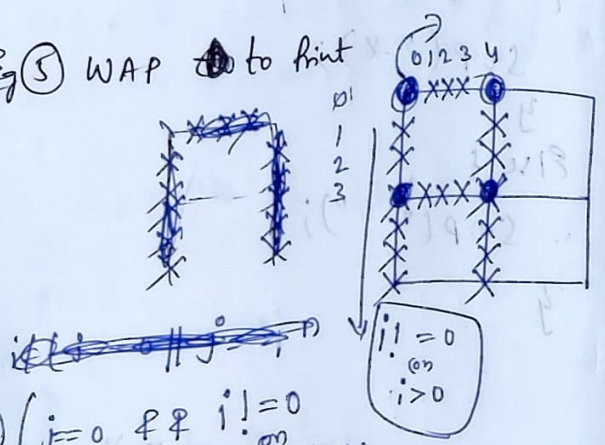
Eg④ WAP to Print

```
i=0 * * * * *
    *       *
    * * * * *
    *       *
    * * * * *
```

```
if(i==0 || j==(n/2) || i==n-1 ||
   j==0)
```



Eg ⑤ WAP to Print

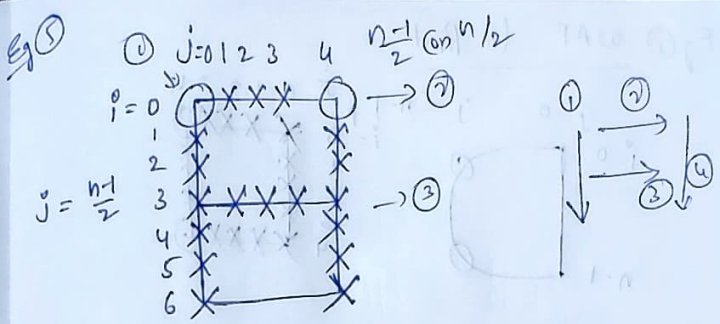


① (j == 0 && i != 0 (or) i > 0) ||

(i == 0 && j > 0 && j < n-1/2) ||

(i == (n-1/2) && j <= n-1/2

Eg ⑤



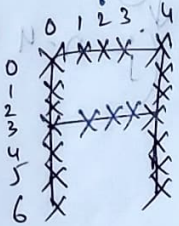
if (j == 0 && i != 9 (or) i > 0) || → ①

(i == 0 && j > 0 && j < (n-1/2)) || → ②

← (i == n-1/2 && j <= n-1/2) ||

③ ④ ← (j == (n-1/2) && i > 0 (or) i != 0)

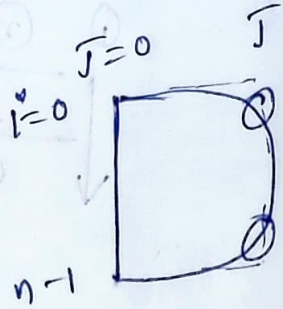
Eg ⑥ WAP to Print

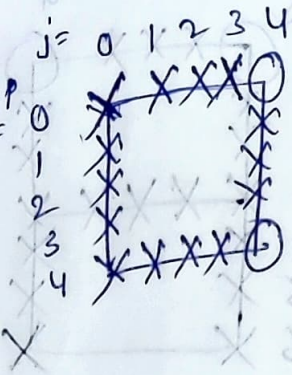


if (i == 0 || j == 0 ||

j == (n-1/2) || j == n-1)

Eg ② WAP to Print

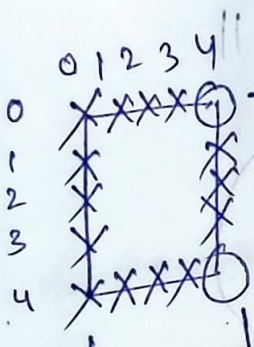




→ if((i==0 && i<n-1)||(j==0)||

(i==n-1 && i<n-1)

||((i==0 && j<n-1)



①→(i==0 && j<n-1)

②→(i==n-1 && i<n-1)

③(j==n-4 && j<n-1 && j>0)