Basic Square patterns (Nested loops)

Q1 print an nxn star square (dynamic input from user)

let's assume n=3

O/P

```
*_*_*_
*_*_*_
*_*_*_
```

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

Dry run

int i=1 [1]

Condition check i≤n

1≤3 → true

Enter into loop

j=1 [1]

condition check j≤n

1≤3 → true

Enter into 2nd loop

kaam → print *

j++ (1+1→2)

j=2 [2]

Condition check j≤n

2≤3 → true

Enter into loop

kaam → print *

j++ (2+1→3)

j=3 [3]

Condition check j≤n

3=3 → true

kaam → print *

j++ (3+1→4)

j=4 [4]

Condition check j≤n

4≤3 → false

2nd loop end

System.out.println(); → next line

i++ (1+1→2)

i=2 [2] i≤n (2≤3) → true

again j=1 [1] → 2nd time 2nd loop

Condition= j≤n (1≤3) → print *-

j++ (1+1→2)

Condition= j≤n → true (2≤3) → print *-

j++ (2+1→3)

Condition: j≤n → (3≤3) → true print *-

j++ (3+1→4)

Condition: j≤n (4≤3) → false

end 2nd loop

System.out.println(); → next line

i++ (2+1→3)

i=3 [3] (i≤n) → (3≤3) → true

again j=1 [1] → 3rd time 2nd loop

Condition: j≤n (1≤3) → true → print *-

j++ (1+1→2)

j=2 [2]
Condition j≤n (2≤3) → true
print *
j++ (2+1→3)
j=3 [3]
Condition j≤n (3=3) → true
print *
j++ (3+1→4)
j=4 [4]
Condition j≤n (4≤3) → false
end 2nd loop
System.out.println(); → next time
i++ (3+1→4)
i=4 [4]
Condition for 1st loop
i≤n (4≤3) – false
end 1st loop

```
import java.util.Scanner;
public class main{
public static void main(String[] args)
{
    Scanner S = new Scanner(System.in);
System.out.println("Enter Number:");
    int n = S.nextInt();
for int i,j;
for(i= 0|; i≤n; i++)
{
for (j=1; j≤n ; j++)
{
System.out.print("* ");
}
System.out.println();
}
}}
```

What does outer loop represent?
outer loop is used to define how many times inner loop execute
means outer loop show ~~enter~~ rows in pattern
because when inner loop end outer loop starts new row using System.out.println();

What does inner loop represent?
Inner loop is used to print '*' start at every iteration of inner loop.
means it define how many columns to print
because n=3.
inner loop run for 3 times at each time print * & when inner loop end next line starts from new line.

Q:2 print the following pattern

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

o/p

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

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

Dry run

int i = 1 [1]

condition check i<=4

1 <= 4 → true

enter into 1st loop

j=1 [1]

j <= 4 (1<=4) → true

Kaam

s.o.print (" "+i) → print value of i [1]

j++ (1+1→2)

j=2 [2]

j<=4 (2<=4) → true

so. print (" "+i) → 1

j++ (2+1→3)

j<=4 (3<=4) → true

j=3 [3]

s.o.print (" "+i) → 1

j++ (3+1→4)

j=4 [4] j<=4 → true

s.o.print (" "+i) → 1

j++ (4+1→5)

j=5 [5] j<=4 (4<=5) → false

loop end

System.out.println(); → next line

i++ (1+1→2)

[2] 2<=4 → true

j=1 [1]

j<=4 → (1<=4) → true

s.o.print (" "+i) → 2

j++ (1+1→2)

j=2 [2] j<=4 → 2<=4 → true

s.o.print (" "+i) → 2

j++ (2+1→3)

j=3 [3] j<=4 → 3<=4 → true

s.o.print (" "+i) → 2

j++ (3+1→4)

j=4 [4] j<=4 → 4<=4 → true

s.o.print (" "+i) → 2

j++ (4+1→5)

j=5 [5] j<=4 → 5<=4 → false

2nd loop end

System.out.println(); → next line

i++ (2+1→3)

i=3 [3] i<=4 (3<=4) → true

again 2nd loop

S.o.print(" "+i) → 4
j++ (4+1 → 5)
j=5 [5] j≤4 (5≤4) → false
2nd loop end
System.out.println();
i++ (4+1 → 5) i=5 [5]
i≤4 (5≤4) → false
1st loop end

```
public class Numsquare{ {
public static void Square (String[] args)
   {
      int i, j;
for (i=1 ; i ≤ 4 ; i++)
   {
      for (j=1 ; j≤4 ; j++)
        {
      System.out.print (" "+i);
        }
      System.out.println();
   }
} }
```

outer loop → use to print number Because for each row number is same so, inner for loop change value for each iteration & outer loop change value when inner loop end & we goes to next line.
inner loop → inner loop represent position of each number as after each inner iteration we goes to next position number remain same.

Q:3 print the following pattern

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

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

Dry run

int i=1 [1]

i<=4 (1<=4) true

enter into 1st loop

j=1 [1]

j<=4 (1<=4) → true

enter into 2nd loop

~~j<=4~~ (S.o.print(" "+j) → 1 o/p 1

j++ (1+1→2) j=2 [2] o/p _1_2

j<=4 (2<=4) → true

S.o.print(" "+j) → 2 o/p _1_2_3

j++ (2+1→3) j=3 [3]

j<=4 (3<=4) → true

S.o.print(" "+j) → 3

j++ (3+1→4) [4] o/p _1_2_3_4

j<=4 (4=<4) → true

S.o. print(" "+j) → 4 o/p _1_2_3_4

j++ (4+1→5) j=5 [5]

j<=4 (5<=4) → false

~~for~~ 2nd loop end

System.out.println() o/p _1_2_3_4

i++ (1+1→2) i=2 [2]

i<=4 (2<=4) → true

j=1 [1]

j<=4 (1<=4) → true

enter into 2nd loop

System.o.print(" "+j) → 1 o/p _1_2_3_4

j++ (1+1→2) ·_1

[2] j<=4 (2<=4) → true

S.o.print(" "+j) → 2 o/p _1_2_3_4

j++ (2+1→3) [3] ·_1_2

j<=4 (3<=4) → true

S.o.print(" "+j) → 3 o/p _1_2_3_4

j++ (3+1→4) [4] _1_2_3

j<=4 (4<=4) → true

System.o.print(" "+j) → 4

j++ (4+1→5) [5]

j<=4 (5<=4) → False

end 2nd loop

i++ (2+1→3) [3]

i<=4 (3<=4) → true

j=1 [1]

j<=4 ☒ (1<=4) → true

enter into 2nd loop again

S.o.print(" "+j) → 1
j++ j=2 [2]
j≤2 → true
S.o.Print(" "+j) → 2
j++ j=3 [3]
j≤3 → true
S.o.print(" "+j) → 3
j++ j=4 [4]
j≤4 (4=4) → true
S.o.print(" "+j) → 4
j++ j=5 [5]
j≤4 (5≤4) → false
end 2nd loop
i++ (3+1) → 4 [4]
enter into 1st loop
j=1 [1]
j≤4 ☐ true
enter into 2nd loop
S.o.print(" "+j) → 1
j++ → [2]
j≤4 → true
S.o.print(" "+j) → 2
j++ (2+1) → 3 [3]
S.o.print(" "+j) → 3
j++ (4) [4]
j≤4 → true
S.o.print(" "+j) → 4
j++ (4+1) → 5 [5]
j≤4 (5≤4) → false (end 2nd loop)
i++ (5≤4) → false (end 1st loop)

```
class Squareofnum{
public static void main (String[] arg)
 {
    int i,j;
 for (i=1; i≤4; i++)
    {
     for (j=1; j≤4; j++)
      {
      System.out.print(" "+j);
      }
     System.out.println();
    }
}
```

outer loop →
use to print number for each row after i++ new row starts
inner loop →
print 'j' means for each iteration value is change so we also want to print 1+1 value for each position.
∴ we print j
also after inner loop end ∴ i condition get satisfy again j=1 to initialize ∴ for new row printing starts from 1.

Nbyn.java

```java
import java.util.Scanner;
public class Nbyn{

public static void main(String[] args){
    Scanner s = new Scanner(System.in);
    System.out.println("Enter Number : ");

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

```
C:\DSA>javac Nbyn.java

C:\DSA>java Nbyn.java
Enter Number :
4
* * * *
* * * *
* * * *
* * * *

C:\DSA>
```

FourHnum.java > Language Support for Java(TM) by Red Hat > F

```java
public class FourHnum{

    public static void main(String[] args) {
        int i,j;
        for(i=1;i<=4;i++){
            for(j=1;j<=4;j++){
                System.out.print(" "+i);
            }
            System.err.println();
        }
    }
}
```

```
C:\DSA>javac FourHnum.java

C:\DSA>java FourHnum.java
 1  1  1  1
 2  2  2  2
 3  3  3  3
 4  4  4  4
```

FourVnum.java > Language Support for Java(TM) by Red Hat > FourV

```java
public class FourVnum {

    public static void main(String[] args) {
        int i,j;
        for(i=1;i<=4;i++){
            for(j=1;j<=4;j++){
                System.err.print(" "+j);
            }
            System.err.println();
        }
    }

}

```

```
C:\DSA>javac FourVnum.java

C:\DSA>java FourVnum.java
 1 2 3 4
 1 2 3 4
 1 2 3 4
 1 2 3 4

C:\DSA>
```