**Pattern using Recursion**

**1. Star using recursion and loop:**

**private** **static** **void** printStar(**int** num, **int** j) {

**if** (j>=num)

**return**;

**else** {

*printStar*(num, j+1);

**for** (**int** i=0; i<=num-j; i++)

System.***out***.print(" ");

**for** (**int** k=0; k<=j;k++)

System.***out***.print("\* ");

System.***out***.println();

}

}

**2. Inverted Star using recursion and loop:**

**private** **static** **void** printStar(**int** num, **int** j) {

**if** (j>=num)

**return**;

**else** {

*printStar*(num, j+1);

**for** (**int** i=0; i<=num-j; i++)

System.***out***.print(" ");

**for** (**int** k=0; k<=j;k++)

System.***out***.print("\* ");

System.***out***.println();

}

}

**3. Inverted Star Without Loop(num=5;j=0):**

**private** **static** **void** PrintStarNoLoop(**int** num, **int** j) {

**if** (num<1)

**return**;

**if** (j<num) {

System.***out***.print("\* ");

*PrintStarNoLoop*(num, j+1);

}

**else** {

System.***out***.println(); *PrintStarNoLoop*(num-1,0);

}

}

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

**OUTPUT:**

\* \* \* \*

\* \* \*

\* \*

\*

**OUTPUT:**

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**4. Print Star Without Loop (j=0; k=0):**

**private** **void** PrintStarNoLoop(**int** num, **int** j, **int** k) {

**if** (j>num-1)

**return**;

**if** (k<=j) {

System.***out***.print("\* ");

*PrintStarNoLoop*(num, j, k+1);

}

**else** {

System.***out***.println(); *PrintStarNoLoop*(num,j+1, 0);

}

}

**5. Star Without Loop(Different method):**

**private** **static** **void** PrintStarNoLoop(**int** num, **int** j) {

**if** (num<1)

**return**;

**if** (j<num) {

*PrintStarNoLoop*(num, j+1);

System.***out***.print("\* ");

}

**else** {

*PrintStarNoLoop*(num-1,0);

System.***out***.println();

}

}

**6. Inverted Triangle Star Without Loop:**

**void** printStarTrianlge(**int** o\_num, **int** num, **int** j, **int** k) {

**if** (num<1)

**return**;

**if** (k-num!=0) {

System.***out***.print(" ");

*printStarTrianlge*(o\_num, num, j, k-1);

}

**else** {

**if** (j<num) {

System.***out***.print("\* ");

*printStarTrianlge*(o\_num, num, j+1, k);

}

**else** {

System.***out***.println();

*printStarTrianlge*(o\_num, num-1,0 , o\_num);

}

}

}

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**OUTPUT:**

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**7. Triangle Star Without Loop:**

**void** printStarTrianlge(**int** o\_num, **int** num, **int** j, **int** k) {

**if** (num<1)

**return**;

**if** (k<num-1) {

System.***out***.print(" ");

*printStarTrianlge*(o\_num, num, j, k+1);

}

**else** {

**if** (j<o\_num-k) {

System.***out***.print("\* ");

*printStarTrianlge*(o\_num, num, j+1, k);

}

**else** {

System.***out***.println();

*printStarTrianlge*(o\_num, num-1,0 , 0); }

}

}

**8 . Full triangle using recursion and loop:**

**private static void printStar(int num, int j) {**

**System.out.println();**

**if (j>=num)**

**return;**

**else {**

**for (int i=0; i<=num-j; i++)**

**System.out.print(" ");**

**for (int k=0; k<j;k++)**

**System.out.print("\* ");**

**printStar(num, j+1);**

**for (int i=0; i<=num-j; i++)**

**System.out.print(" ");**

**for (int k=0; k<j;k++)**

**System.out.print("\* ");**

**}**

**System.out.println();**

**}**

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*