@github

Pattern Programs

SquarePattern

* * * * Here, we want Row and Col. --- Logic, Nested for-loop:

* * * * 1st loop ---- int i = 1; i <= numberOfRows; i++

* * * * 2nd loop ---- int j = 1; j <= numberOfCols; j++)

* * * * Outer give extra sysout statement. All this give our exapted result.

```
package pattern.special;

class SquarePattern {
   public static void main(String[] args) {
      int numberOfRows = 4;
      int numberOfCols = 4;

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

DecrementPattern

* * * * Here, we want only rows. ---- Logic, Nested for-loop:

* * * 1st loop ---- int i = 1; i <= numberOfRows; i++

* * 2nd loop ---- int j = i; i <= numberOfRows; j++

* Outer give extra sysout statement. All this give our exapted result.

```
package pattern.special;
```

```
public class DecrementPattern {
   public static void main(String[] args) {
      int numberOfRows = 4;

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

IncrementPattern

```
*          Here, we want only rows.      ---- Logic, Nested for-loop:
* *        1st loop ---- int i = 1; i <= numberOfRows; i++
* * *      2nd loop ---- int j = i; i <= i ; j++
* * * *    Outer give extra sysout statement. All this give our exapted result.
```

```
package pattern.special;

public class IncrementPattern {
   public static void main(String[] args) {
      int numberOfRows = 4;

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

```
    }
}
```

DecreIncrePattern

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

Here, we want only rows. ---- Logic, Nested for-loop:

we use above to two programs write in one program, that give the our exapted output.

Try it yourself.

Output:

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac SquarePattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac SquarePattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac DecrementPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac IncrementPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac DecreIncrePattern.java
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.SquarePattern
* * * *
* * * *
* * * *
* * * *
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.DecrementPattern
* * * *
* * *
* *
*
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.IncrementPattern
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.IncrementPattern
*
* *
* *
* *
Focus folder in explorer (ctrl + click)
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.DecreIncrePattern
* * * *
* * *
* *
*
*
* *
* * *
* * * *
```

@github

LowerTriangularPattern

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

Here, we want only rows. ---- Logic, Nested for-loop:

1st loop --- int i = 1; i <= numberOfRows; i++

2nd loop --- int spaces = 1; spaces <= numberOfRows - i; spaces++

3rd loop --- int j = 1; j <= i; j++

```
package pattern.special;

public class LowerTriangularPattern {
   public static void main(String[] args) {
      int numberOfRows = 4;

      for(int i = 1; i <= numberOfRows; i++) {
         for(int spaces = 1; spaces <= numberOfRows - i; spaces++) {
            System.out.print(" ");
         }
         for (int j = 1; j <= i; j++) {
            System.out.print("* ");
         }

         System.out.println();
      }
   }
}
```

UpperTrigularPattern

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

Here, we want only rows. ---- Logic, Nested for-loop:

1st loop --- int i = numberOfRows; i >= 1; i--

2nd loop --- int spaces = 1; spaces <= numberOfRows - i; spaces++

3rd loop --- int j = 1; j <= i; j++

```
package pattern.special;
```

```
public class UpperTriangularPattern {

    public static void main(String[] args) {
        int numberOfRows = 4;

        for (int i = numberOfRows; i >= 1; i--) {
            for (int spaces = 1; spaces <= numberOfRows - i; spaces++) {
                System.out.print(" ");
            }

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

            System.out.println();
        }
    }
}
```

JoinPyramidPattern

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

Here, we want only rows. ---- Logic, Nested for-loop: we use above to two programs write in one program, that give the our exapted output.

ReversePyramidPattern

Here, we want only rows. ---- Logic, nested for-loop: we use above to two programs write in one program, that give the our exapted output.

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

PyramidPattern

Here, we want only rows. ---- Logic, nested for-loop:

To your own to find where code modified to get this output check and comment me.

```
package pattern.special;

public class PyramidPattern {
    public static void main(String[] args) {
        int numberOfRows = 4;

        for (int i = numberOfRows; i >= 1; i--) {
            for (int spaces = 1; spaces <= numberOfRows - i; spaces++) {
                System.out.print(" ");
            }

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

            System.out.println();
        }
```

```
        for(int i = 2; i <= numberOfRows; i++) {
            for(int spaces = 1; spaces <= numberOfRows - i; spaces++) {
                System.out.print(" ");
            }
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }

            System.out.println();
        }

    }
}
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac LowerTriangularPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac UpperTriangularPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac JoinPyramidPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac ReversePyramidPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac PyramidPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special>
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.LowerTriangularPattern
    *
   * *
  * * *
 * * * *
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.UpperTriangularPattern
* * * *
 * * *
  * *
   *
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.JoinPyramidPattern
    *
   * *
  * * *
 * * * *
  * * *
   * *
    *
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.ReversePyramidPattern
 * * * *
  * * *
   * *
    *
    *
   * *
  * * *
 * * * *
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.special.PyramidPattern
 * * * *
  * * *
   * *
    *
   * *
  * * *
 * * * *
PS C:\ayekiran\DSA\critical-thinking\java\small\src>
```

ReverseDecrementPattern

**** Here, we want only rows. ---- Logic, Nested for-loop:
 *** 1st loop --- int i = 1; i <= numberOfRows; i++
  ** 2nd loop --- int spaces = 0; spaces < i; spaces++
   * 3rd loop --- int j = numberOfRows; i <= j; j—

```
package pattern.special;

public class ReverseDecrementPattern {
   public static void main(String[] args) {
      int numberOfRows = 4;

      for(int i = 1; i <= numberOfRows; i++) {
         for (int spaces = 0; spaces < i; spaces++) {
            System.out.print(" ");
         }

         for (int j = numberOfRows; i <= j; j--) {
            System.out.print("*");
         }
         System.out.println();
      }
   }
}
```

ReverseIncrementPattern

   * Here, we want only rows. ---- Logic, Nested for-loop:
  ** 1st loop --- int i = 1; i <= numberOfRows; i++
 *** 2nd loop --- int spaces = numberOfRows; spaces > i; spaces--
**** 3rd loop --- int j = 1; j <= i; j++

```
package pattern.special;
```

```
public class ReverseIncrementPattern {
    public static void main(String[] args) {
        int numberOfRows = 4;

        for (int i = 1; i <= numberOfRows; i++) {
            for (int spaces = numberOfRows; spaces > i; spaces--) {
                System.out.print(" ");
            }

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

ReverseIncreDecrementPattern

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

Here, we want only rows. ---- Logic, nested for-loop:

To your own to find where code modified to get this output check and comment me.

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac ReverseDecrementPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac ReverseIncrementPattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special> javac ReverseIncreDecrePattern.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\special>
```