Pattern Programs

SquarePattern

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

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.number;

public class Pattern1111 {
   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(i);
         }
         System.out.println();
      }
   }
}
```

1234
1234
1234
1234 --- Try this just change one small change

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

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. Pay cresol role.

```
package pattern.number;

public class Pattern1234Continue {
    public static void main(String[] args) {
        int numberOfRows = 4;
        int numberOfCols = 4;
        int number = 1;

        for (int i = 1; i <= numberOfRows; i++) {
            for (int j = 1; j <= numberOfCols; j++) {
                System.out.printf("%3d ", number);
                number++;
            }
            System.out.println();
        }
    }
}
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac Pattern1111.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac Pattern1234.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac Pattern1234Continue.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number>
```

Try different pattern using above logic but change basic this gets out lot of patterns.

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.Pattern1111
1111
2222
3333
4444
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.Pattern1234
1234
1234
1234
1234
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.Pattern1234Continue
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 14 15 16
PS C:\ayekiran\DSA\critical-thinking\java\small\src>
```

TriangularPattern

1
12
123
1234

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

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

2nd loop ---- int j = 1; j <= i; j++

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

```
package pattern.number;

public class LowerTriangular {
    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(j);
            }
            System.out.println();
        }
    }
}
```

1234 Here, we want Row and Col. --- Logic, Nested for-loop:

234 1st loop ---- int i = 1; i <= numberOfRows; i++

34 2nd loop ---- int j = i; j <= numberOfRows; j++

4 Outer give extra sysout statement. All this give our exapted result. Try this youself.

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac LowerTriangular.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac UpperTriangular.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number>
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.LowerTriangular
1
12
123
1234
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.UpperTriangular
1234
234
34
4
PS C:\ayekiran\DSA\critical-thinking\java\small\src>
```

ReverseTrianglePattern

1234 Here, we want Row and Col. --- Logic, Nested for-loop:

123 1st loop ---- int i = numberOfRows; i >= 1; i--

12 2nd loop ---- int spaces = 0; spaces < numberOfRows - i; spaces++

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

```
package pattern.number;

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

        for(int i = numberOfRows; i >= 1; i--) {
            for (int spaces = 0; spaces < numberOfRows - i; spaces++) {
```

```
            System.out.print(" ");
        }

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

1 Here, we want Row and Col. --- Logic, Nested for-loop:
12 1st loop ---- int i = 1; i <= numberOfRows; i++
123 2nd loop ---- int spaces = 0; spaces < numberOfRows - i; spaces++
1234 3rd loop ---- int j = 1; j <= i; j++ **Try yourself**

1234 Here, we want Row and Col. --- Logic, Nested for-loop:
123 1st loop ---- int i = 1; i <= numberOfRows; i++
12 2nd loop ---- int spaces = 0; spaces < numberOfRows - i; spaces++
1 3rd loop ---- int j = 1; j <= i; j++ By using above two programs.
12
123
1234

```
package pattern.number;

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

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

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

        for(int i = 2; i <= numberOfRows; i++) {
            for (int spaces = 0; spaces < numberOfRows - i; spaces++) {
                System.out.print(" ");
            }

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

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac ReverseUpperTriangle.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac ReverseLowerTriangle.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac FullReverseUpperLowerTriangle.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number>
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.ReverseUpperTriangle
1234
 123
  12
   1
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.ReverseLowerTriangle
   1
  12
 123
1234
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.FullReverseUpperLowerTriangle
1234
 123
  12
   1
  12
 123
1234
PS C:\ayekiran\DSA\critical-thinking\java\small\src>
```

PyramidPattern

4 4 4 4
 3 3 3
  2 2
   1

Here, we want Row and Col. --- 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.number;

public class LowerPyramid {
    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(i + " ");
            }
```

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

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

Here, we want Row and Col. --- 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++ Please Try yourself.

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

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

Both above programs help to develop, this structure.

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

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac UpperPyramid.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac LowerPyramid.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac FullPyramid.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number> javac StarPyramid.java
PS C:\ayekiran\DSA\critical-thinking\java\small\src\pattern\number>
```

```
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.LowerPyramid
4 4 4 4
 3 3 3
  2 2
   1
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.UpperPyramid
   1
  2 2
 3 3 3
4 4 4 4
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.FullPyramid
4 4 4 4
 3 3 3
  2 2
   1
  2 2
 3 3 3
4 4 4 4
PS C:\ayekiran\DSA\critical-thinking\java\small\src> java pattern.number.StarPyramid
   1
  2 2
 3 3 3
4 4 4 4
 1 2 3
  1 2
   1
PS C:\ayekiran\DSA\critical-thinking\java\small\src>
```