1. 1 11 121 1331 14641

import java.util.Scanner;

public class Righttriangle

{

public static void main(String[] args)

{

int rows = 6, coef = 1;

for(int i = 0; i < rows; i++)

{

for(int space = 1; space < rows - i; ++space)

{

System.out.print(" ");

}

for(int j = 0; j <= i; j++)

{

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

coef = 1;

else

coef = coef \* (i - j + 1) / j;

System.out.printf("%4d", coef);

}

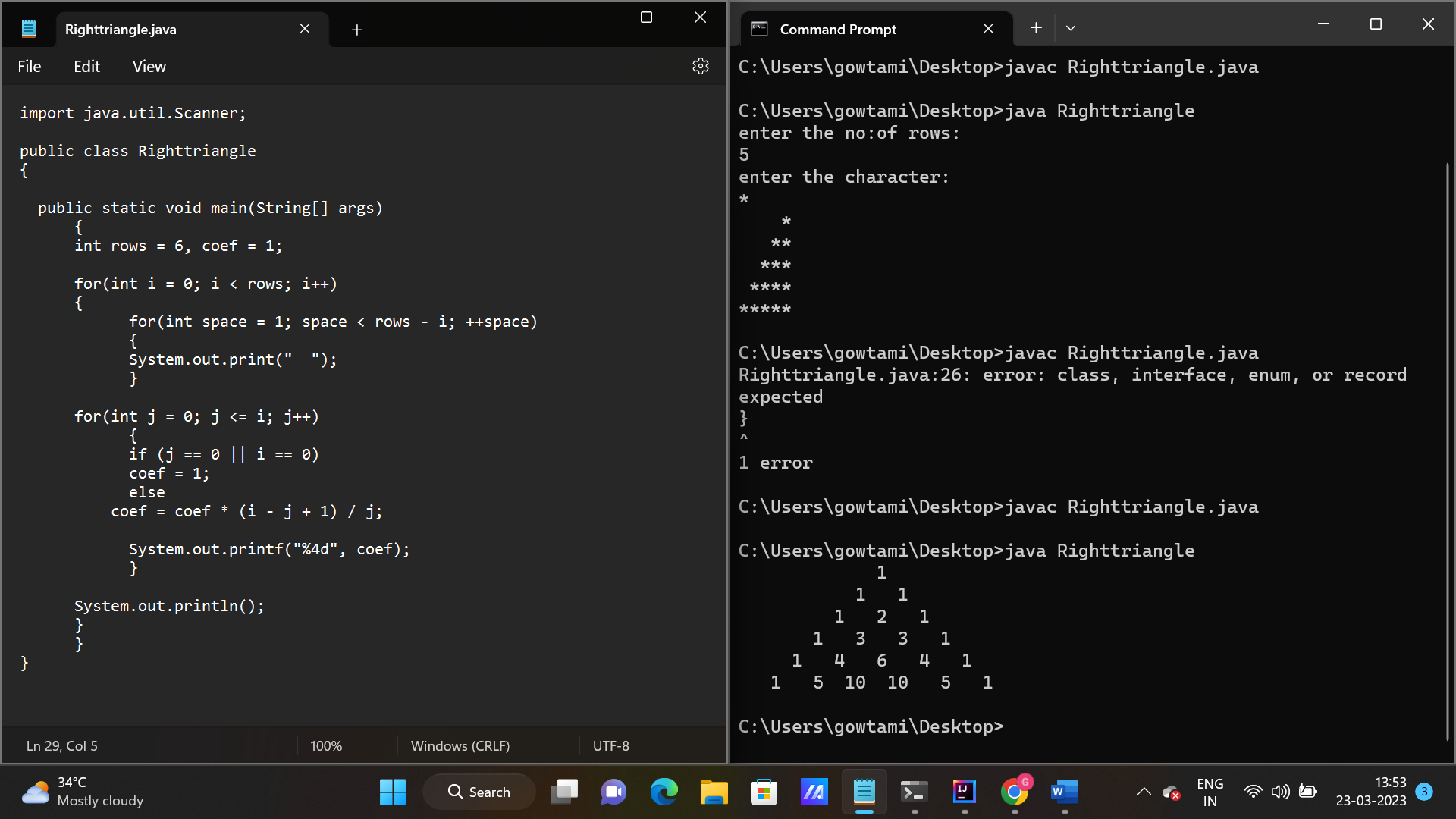
System.out.println();

}

}

}

Output:-



2.full tringle of star:

import java.util.Scanner;

class FullTrianglePattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int n = scanner.nextInt();

for (int i = 1; i <= n; i++) {

// print spaces

for (int j = 1; j <= n - i; j++) {

System.out.print(" ");

}

// print asterisks

for (int k = 1; k <= i; k++) {

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

}

// move to the next line

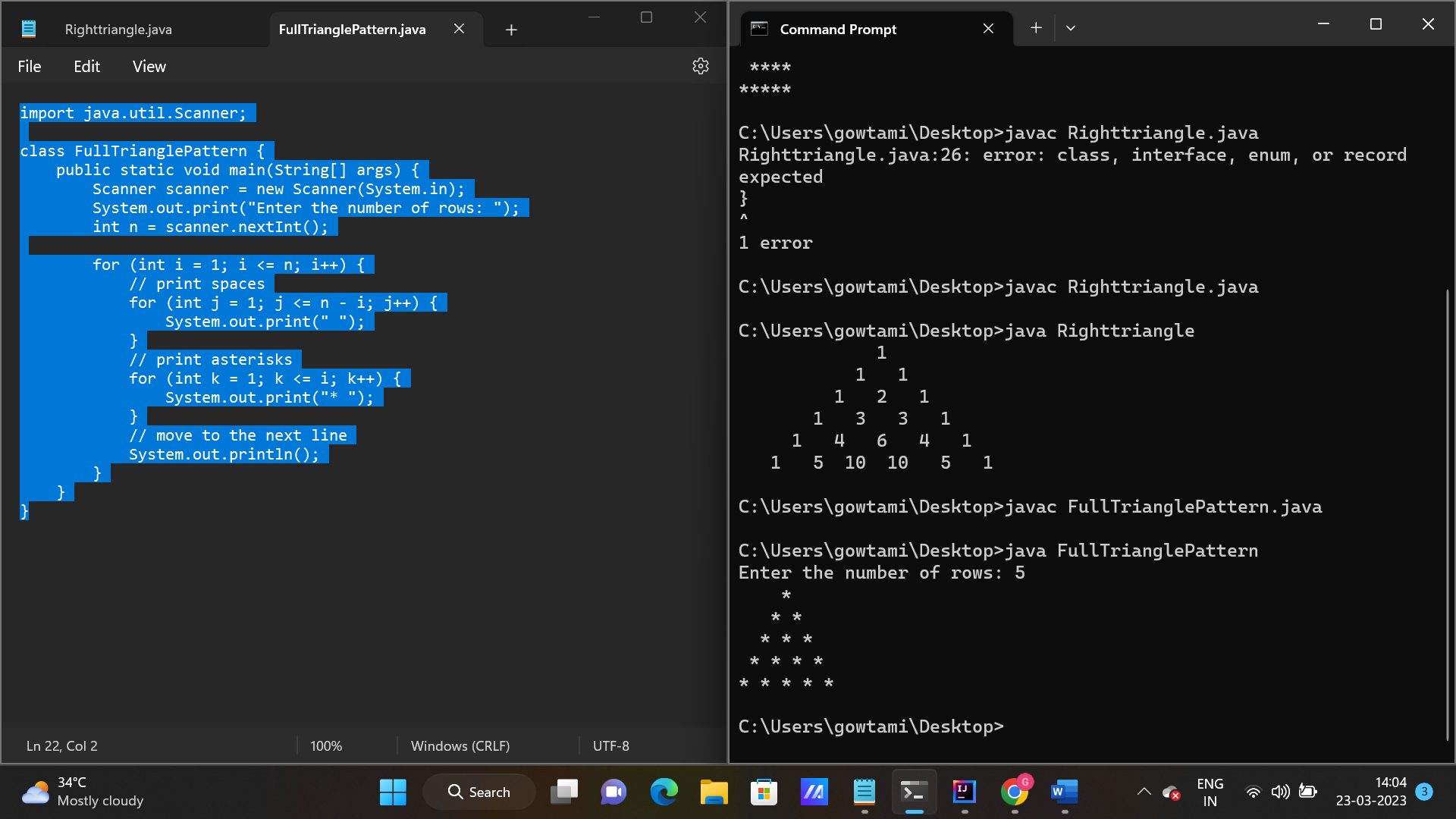
System.out.println();

}

}

}

Output:



3.full rectangle pattern:

import java.util.Scanner;

class RectangularPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int rows = scanner.nextInt();

System.out.print("Enter the number of columns: ");

int columns = scanner.nextInt();

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= columns; j++) {

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

}

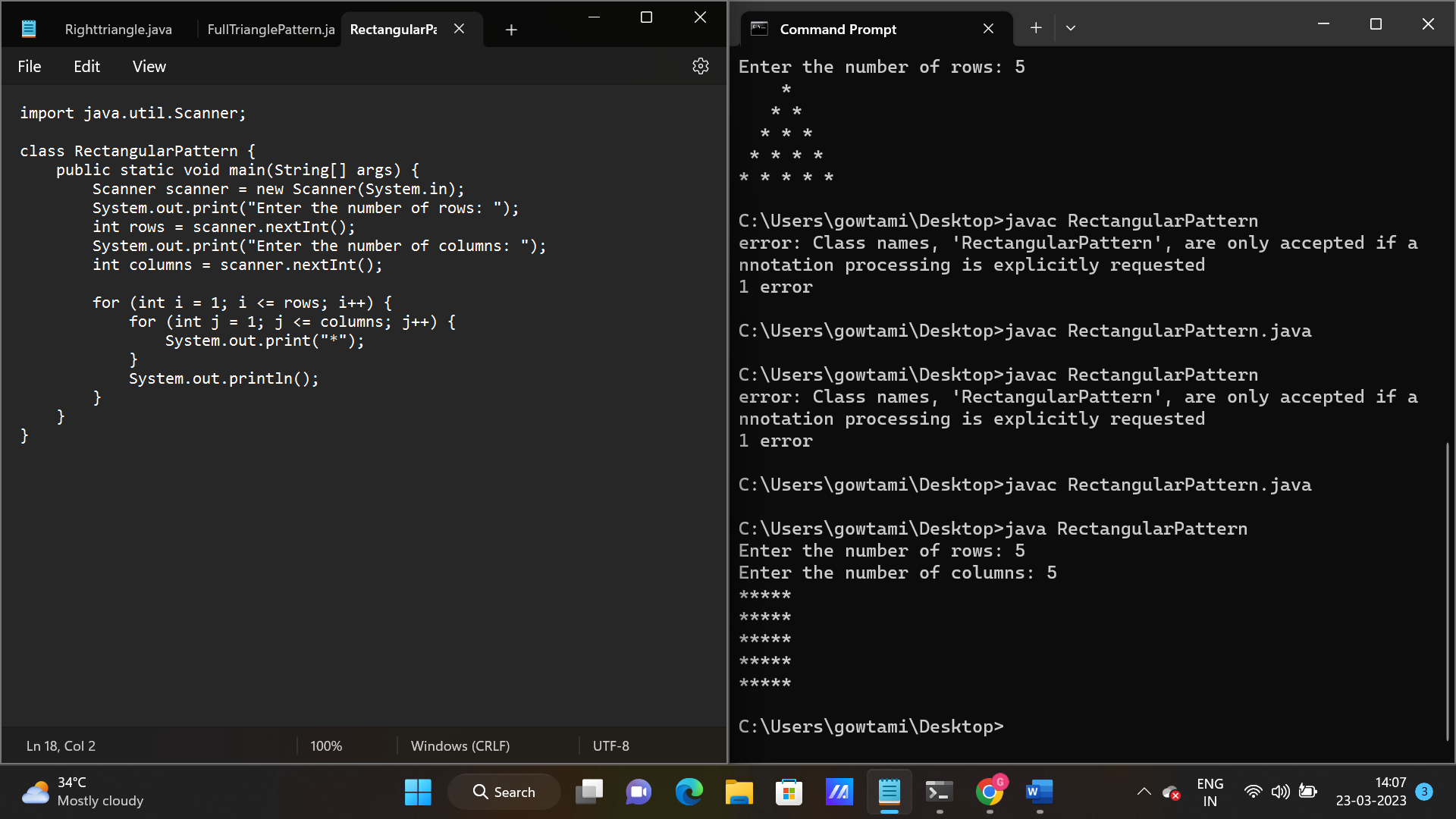
System.out.println();

}

}

}

Output:



4. 1 11 111 pattern

class Pattern {

public static void main(String[] args) {

int n = 3;

// Upper half of the pattern

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(1);

}

System.out.println();

}

// Lower half of the pattern

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= i; j++) {

System.out.print(1);

}

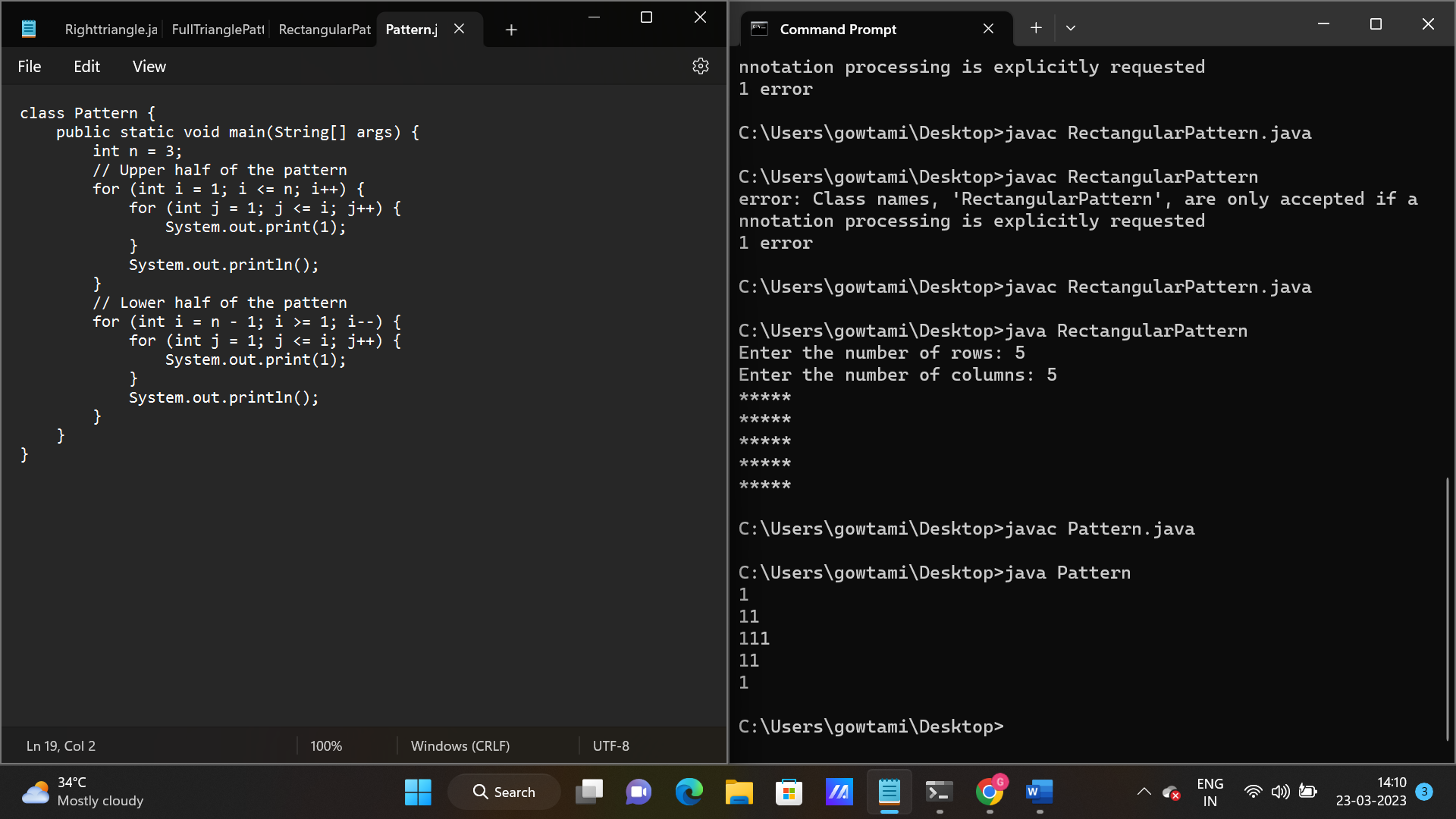
System.out.println();

}

}

}

Output:



5.inverted full triangle pattern

import java.util.Scanner;

class InvertedFullPyramid {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int rows = scanner.nextInt();

for (int i = rows; i >= 1; i--) {

for (int j = i; j < rows; j++) {

System.out.print(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

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

}

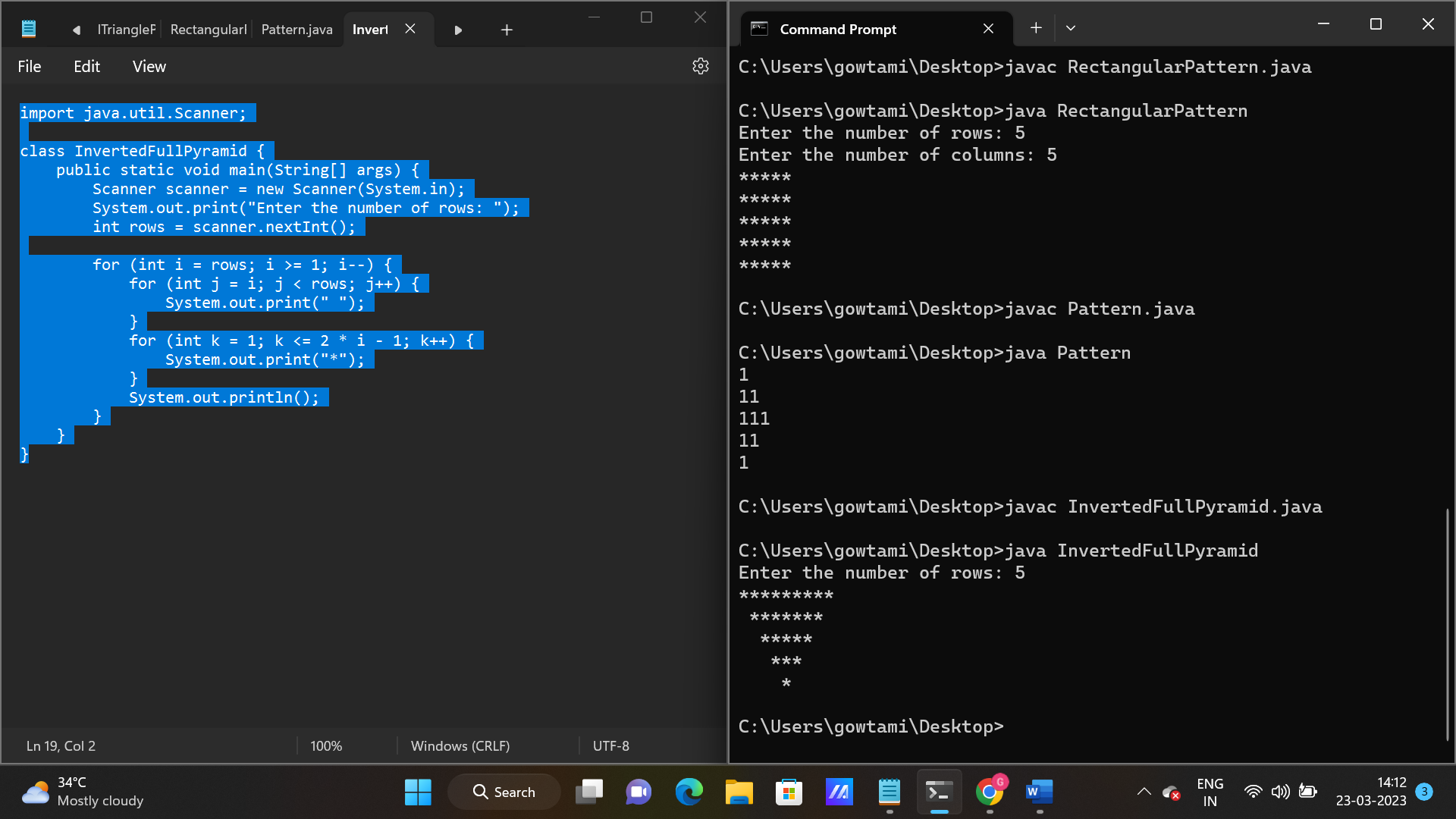
System.out.println();

}

}

}

Output:



6. % symbol pattern

import java.io.\*;

import java.util.\*;

class SymbolPattern

{

public static void main(String args[])

{

try

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number:");

int n=sc.nextInt();

for(int i=1;i<=n;i++)

{

for(int j=1;j<=i;j++)

{

System.out.print("%");

}

System.out.println();

}

}

catch(Exception e)

{

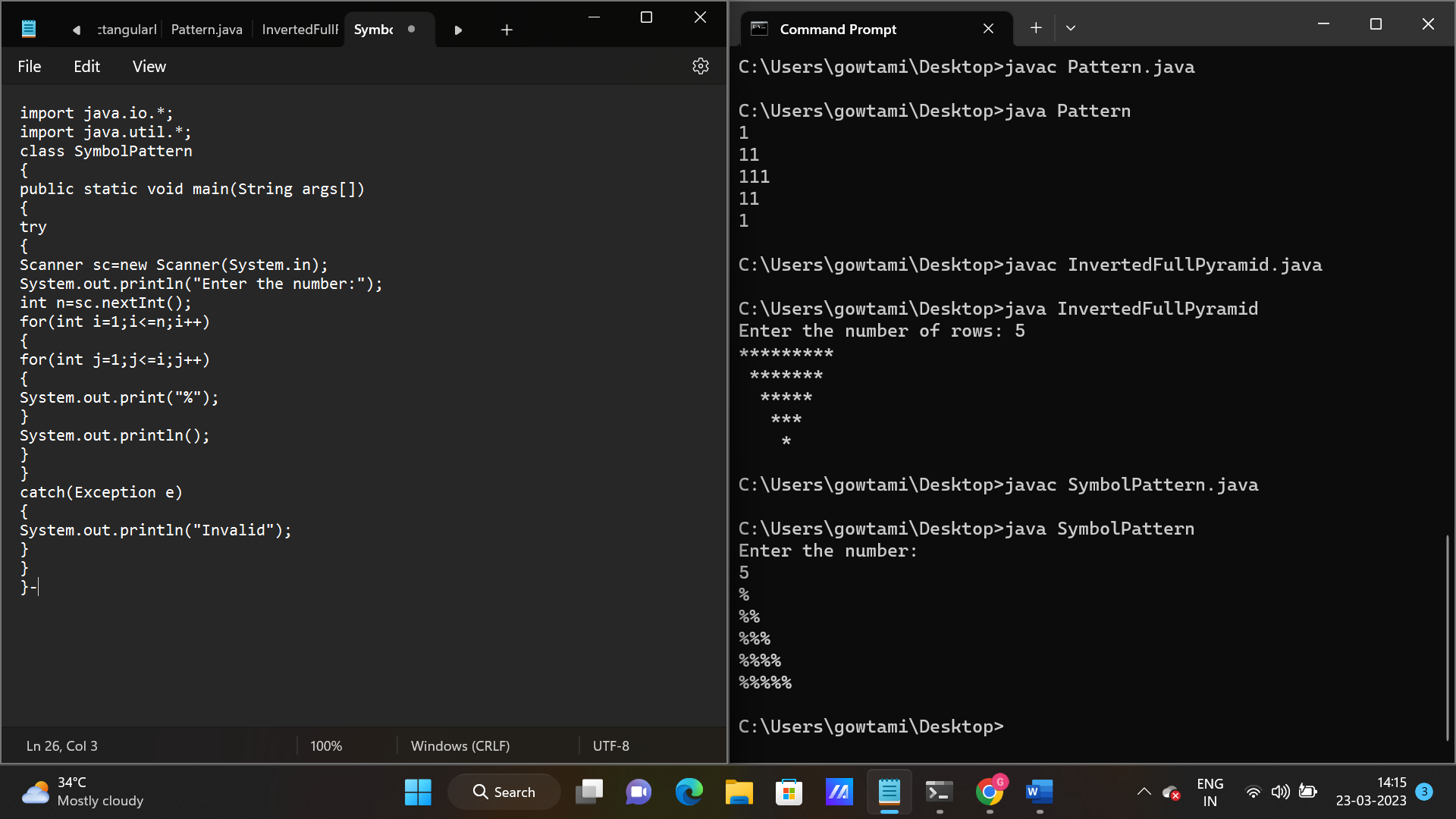
System.out.println("Invalid");

}

}

}

Output:-



7. Hollow square Symbol:

class HollowSquarePattern {

public static void main(String[] args) {

int size = 5;

// Print top row

for (int i = 0; i < size; i++) {

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

}

System.out.println();

// Print middle rows

for (int i = 0; i < size - 2; i++) {

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

for (int j = 0; j < size - 2; j++) {

System.out.print(" ");

}

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

}

// Print bottom row

for (int i = 0; i < size; i++) {

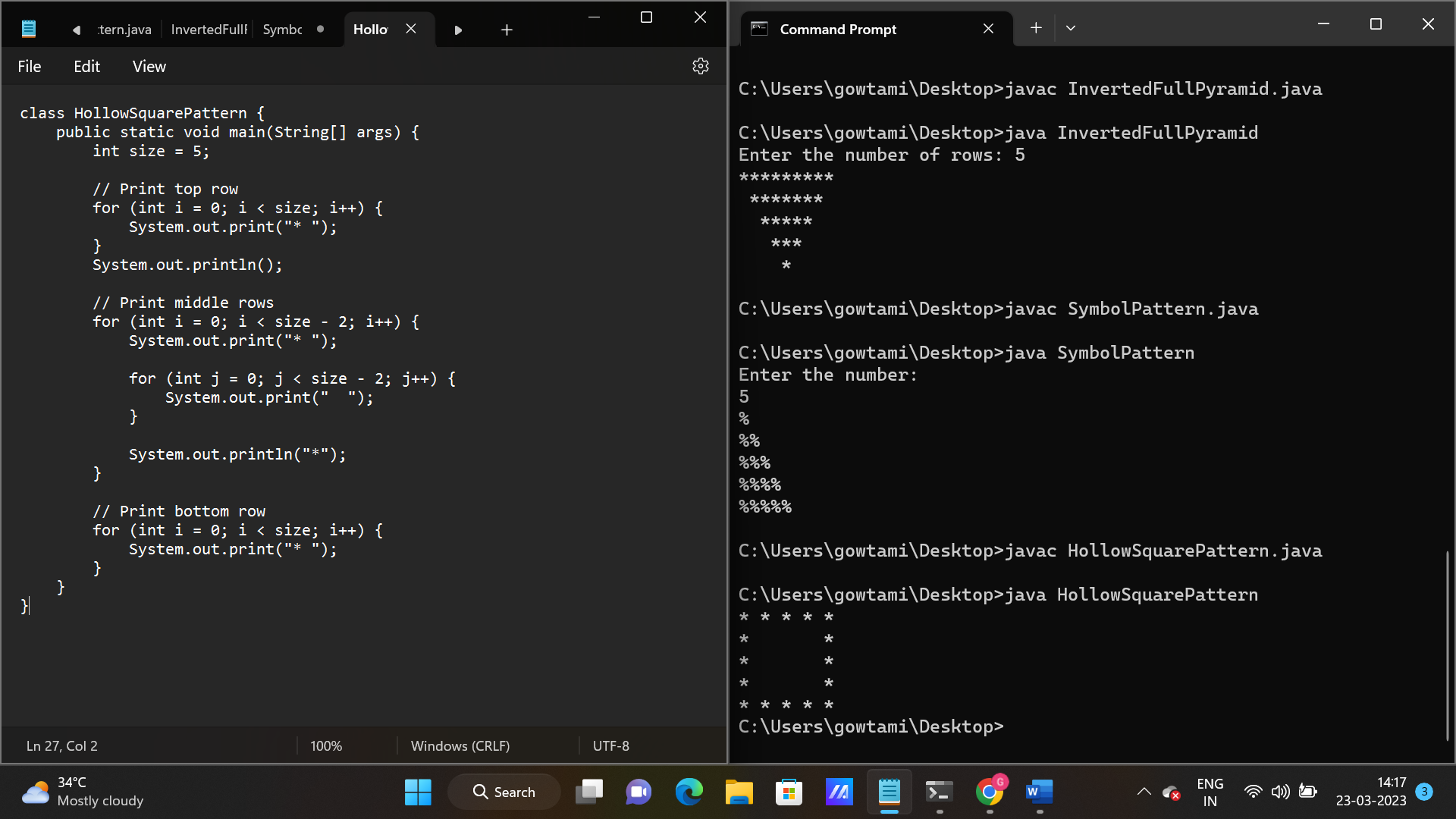
System.out.print("\* ");

}

}

}

Output:-



8. 1 22 333 444 pyramid:

class Pattern122333 {

public static void main(String[] args) {

int n = 4;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(i);

}

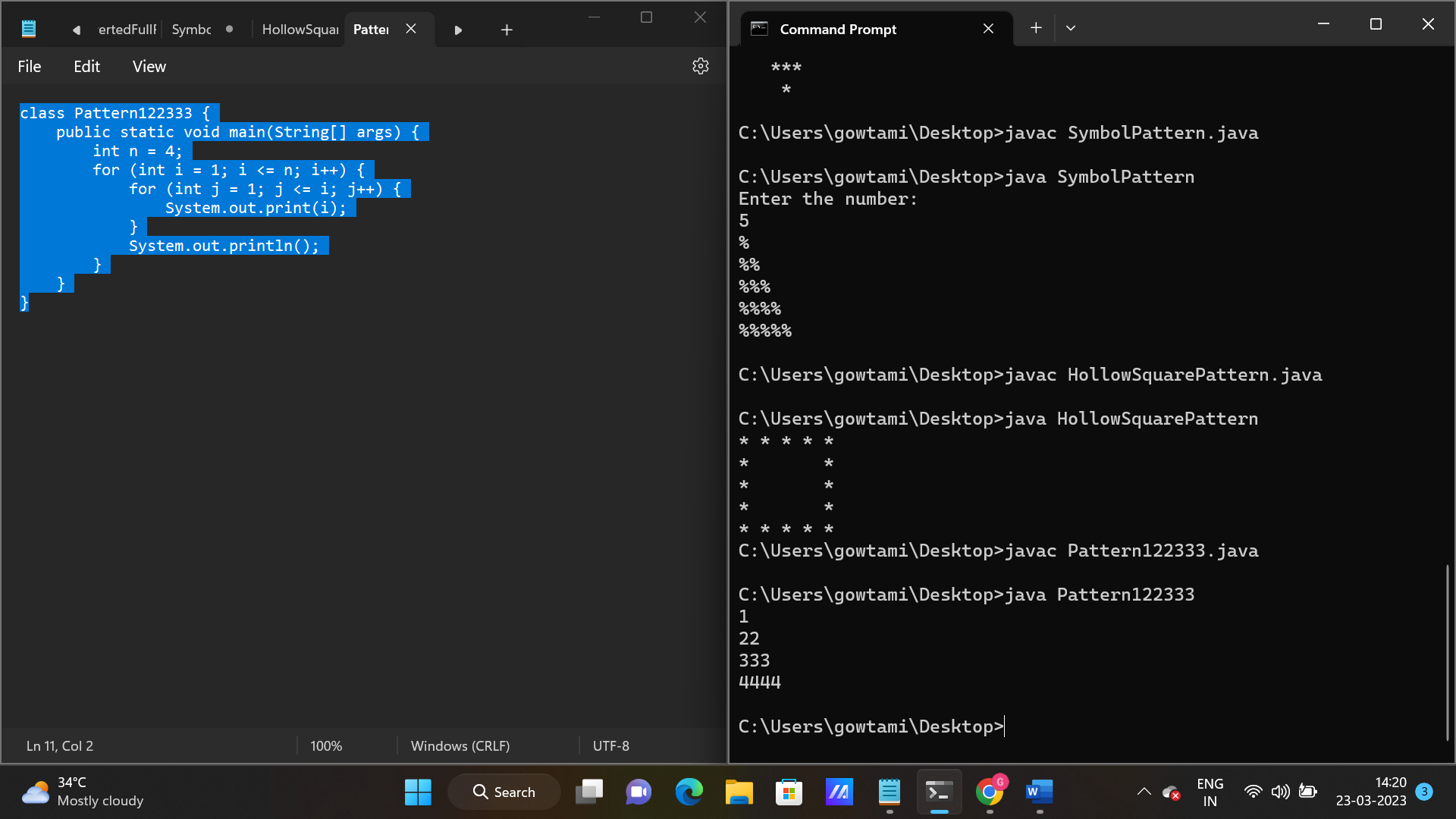
System.out.println();

}

}

}

Output:-



9. 1 4 9 16 pattern pyramid:-

class SqNumberPattern {

public static void main(String[] args) {

int rows = 4;

int num = 1;

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= i; j++) {

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

num++;

}

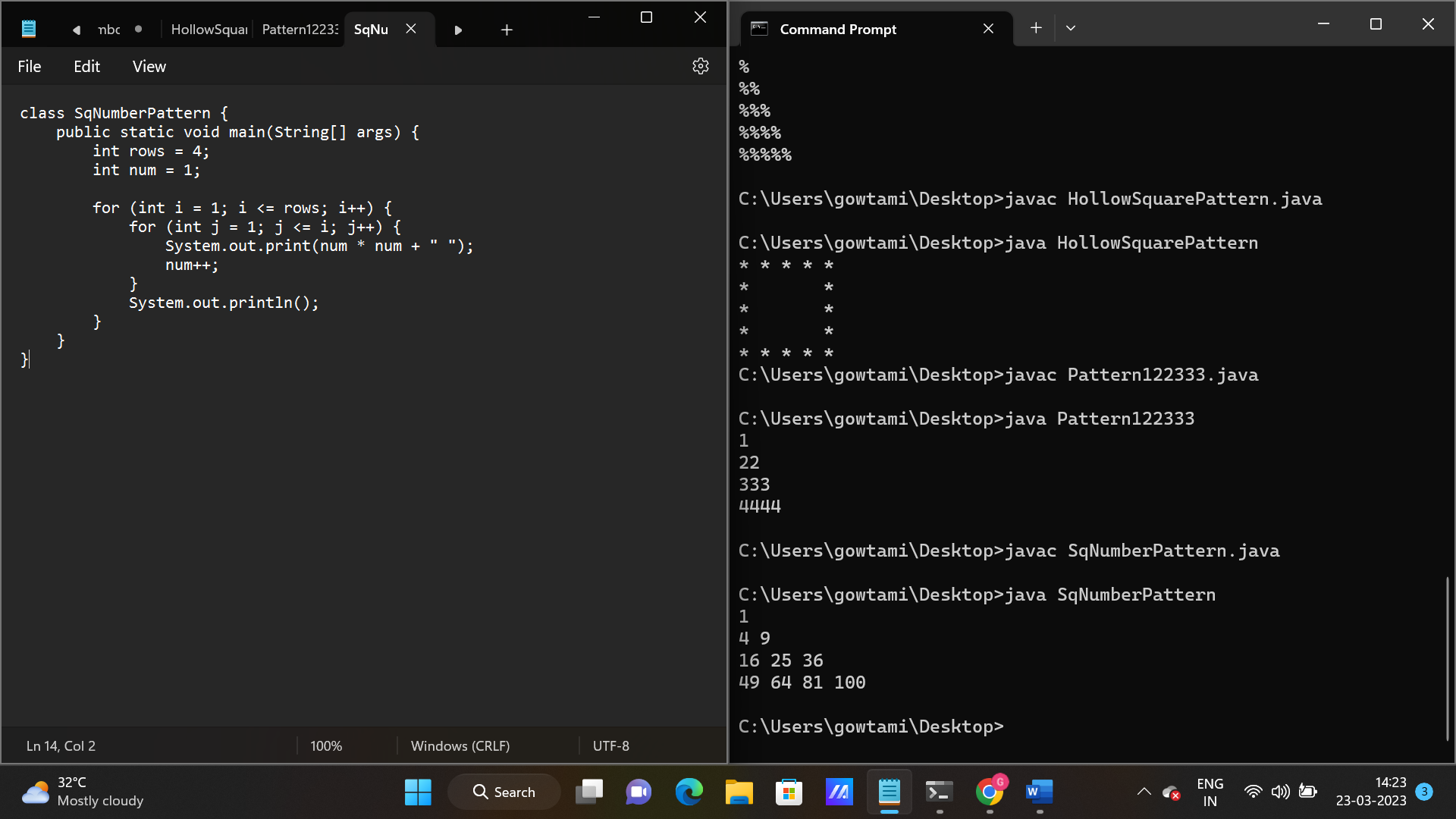
System.out.println();

}

}

}

Output:-



10. FullPatternPyramid1223334444333221

class FullPatternPyramid1223334444333221 {

public static void main(String[] args) {

int n = 4;

// Upper half of the pattern

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(i);

}

System.out.println();

}

// Lower half of the pattern

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= i; j++) {

System.out.print(i);

}

System.out.println();

}

}

}

Output:-

