**Assignment 4**

Printpattern:

1)

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

2)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

3)

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

4)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

5)

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

6)

abcdef

ghijkl

mnopqr

stuvwx

7)

12345

ABCDE

678910

FGHIJ

8)

0

0

0

0

0

9)

0 0

0 0

0

0 0

0 0

10)

1

0

1

0

1

0

11)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

11)

54321

4321

321

21

1

12)

0

101

01010

1010101

**. Rectangle:**

public class RectanglePattern {

public static void main(String[] args) {

int rows = 4, cols = 4;

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

for (int j = 0; j < cols; j++) {

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

}

System.out.println();

}

}

}

**2. Right-angled triangle:**

public class RightAngledTriangle {

public static void main(String[] args) {

int rows = 5;

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

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

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

}

System.out.println();

}

}

}

**3. Inverted right-angled triangle:**

public class InvertedRightAngledTriangle {

public static void main(String[] args) {

int rows = 5;

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

for (int j = rows - i; j > 0; j--) {

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

}

System.out.println();

}

}

}

**4. Right-angled triangle (inverted):**

public class RightAngledTriangleInverted {

public static void main(String[] args) {

int rows = 5;

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

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

System.out.print(" ");

}

for (int j = rows - i; j > 0; j--) {

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

}

System.out.println();

}

}

}

**5. Inverted right-angled triangle (inverted):**

public class InvertedRightAngledTriangleInverted {

public static void main(String[] args) {

int rows = 5;

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

for (int j = rows - i - 1; j >= 0; j--) {

System.out.print(" ");

}

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

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

}

System.out.println();

}

}

}

**6. Alphabetical pattern:**

public class AlphabeticalPattern {

public static void main(String[] args) {

int rows = 4;

char ch = 'a';

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

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

System.out.print(ch++);

}

System.out.println();

}

}

}

**7. Number and Alphabetical pattern:**

public class NumberAlphabeticalPattern {

public static void main(String[] args) {

int rows = 4;

int num = 1;

char ch = 'A';

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

for (int j = 0; j < rows; j++) {

if (i % 2 == 0) {

System.out.print(num++);

} else {

System.out.print(ch++);

}

}

System.out.println();

}

}

}

**8. Vertical zero pattern:**

public class VerticalZeroPattern {

public static void main(String[] args) {

int rows = 5;

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

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

System.out.print(" ");

}

System.out.println("0");

}

}

}

**9. Horizontal and vertical zero pattern:**

public class HorizontalVerticalZeroPattern {

public static void main(String[] args) {

int rows = 5;

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

for (int j = 0; j < rows; j++) {

if (i % 2 == j % 2) {

System.out.print("0 ");

} else {

System.out.print(" ");

}

}

System.out.println();

}

}

}

**10. Alternating 1 and 0 pattern:**

public class AlternatingOneZeroPattern {

public static void main(String[] args) {

int rows = 6;

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

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

System.out.print(" ");

}

System.out.println(i % 2 == 0 ? 1 : 0);

}

}

}

**11. Diamond pattern:**

public class DiamondPattern {

public static void main(String[] args) {

int rows = 5;

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

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

System.out.print(" ");

}

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

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

}

System.out.println();

}

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

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

System.out.print(" ");

}

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

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

}

System.out.println();

}

}

}

**12. Number pyramid:**

Java

public class NumberPyramid {

public static void main(String[] args) {

int rows = 5;

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

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

System.out.print(" ");

}

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

System.out.print(j);

}

for (int j = 2; j <= i; j++) {

System.out.print(j);

}

System.out.println();

}

}

}