

7Loops_And_Pattern2

SESSION WILL START AT 10:00 AM

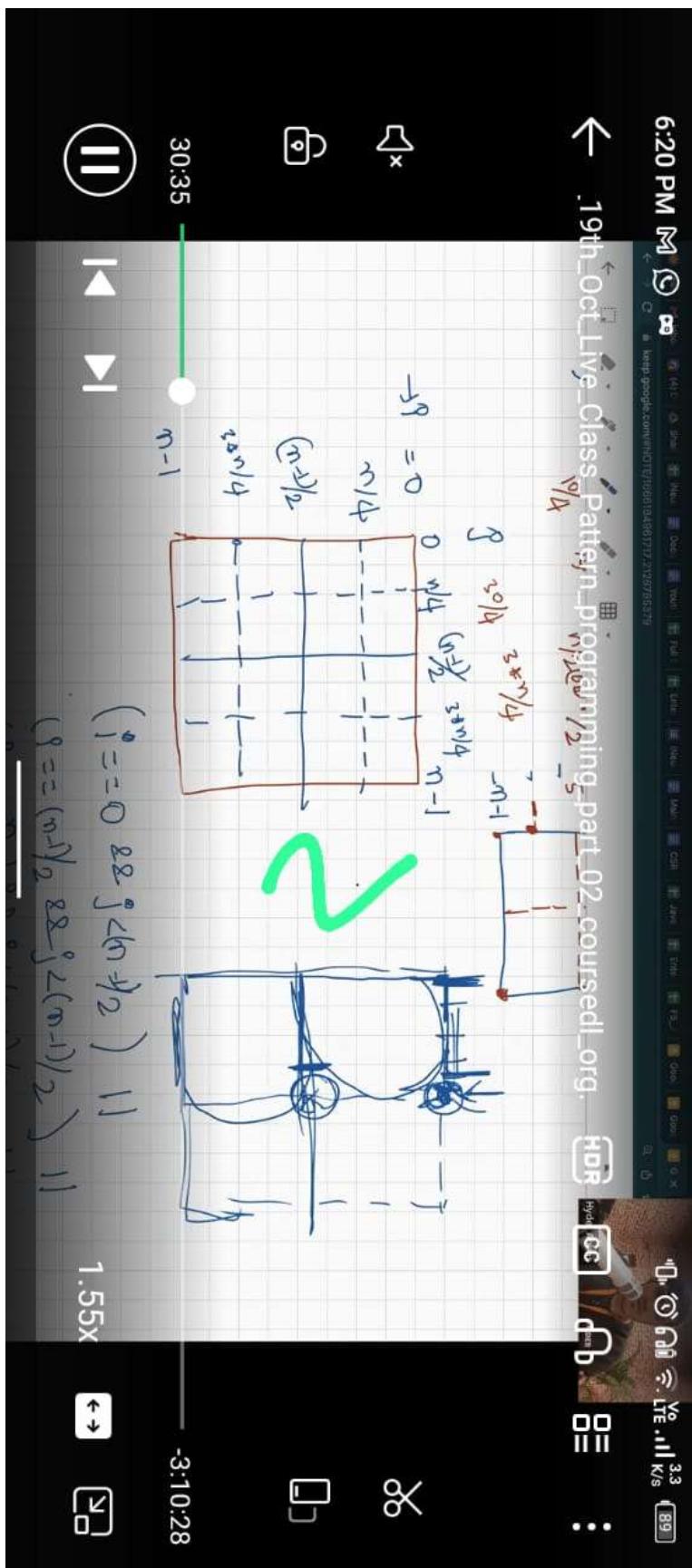
⇒ pattern prog ⇒ To used to programming & thinking

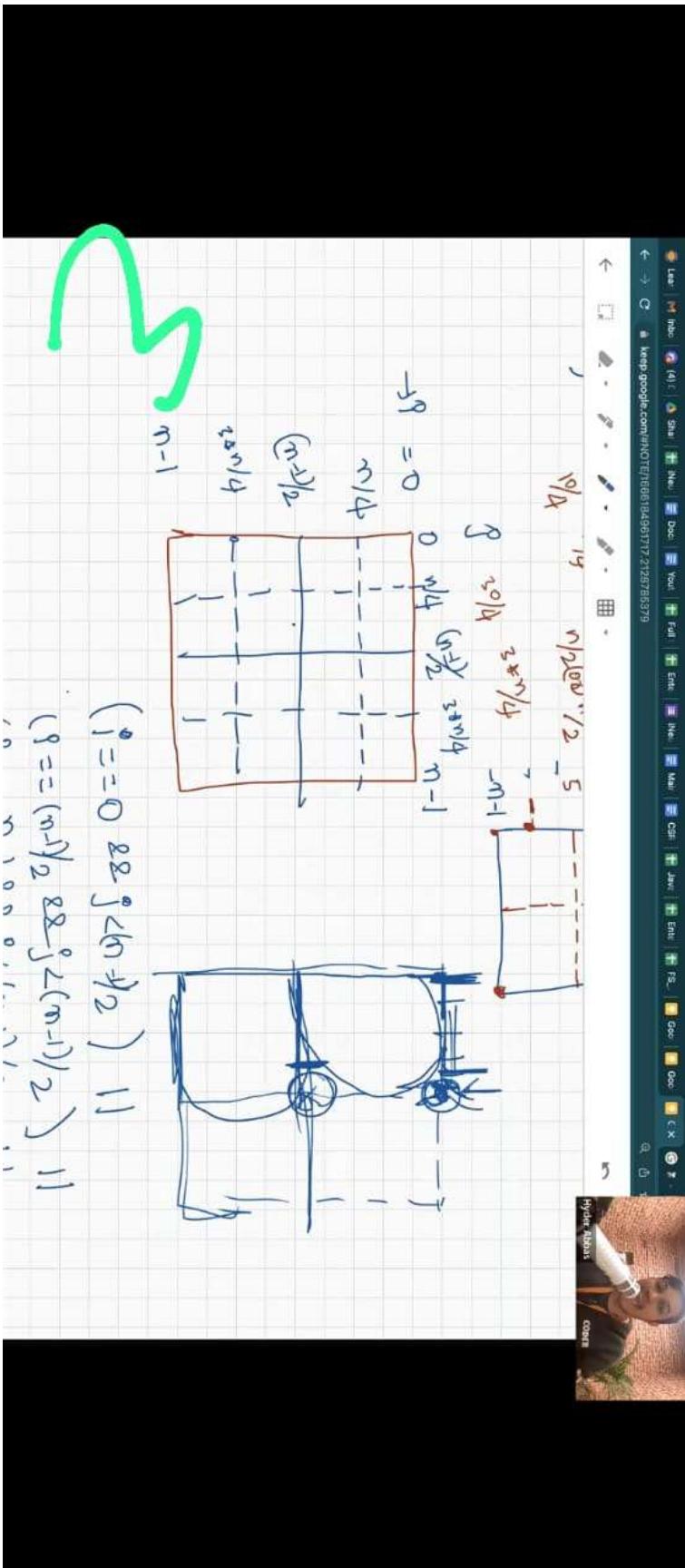
⇒ TONTO ⇒ class & object, JVM Dataarea, instances -----

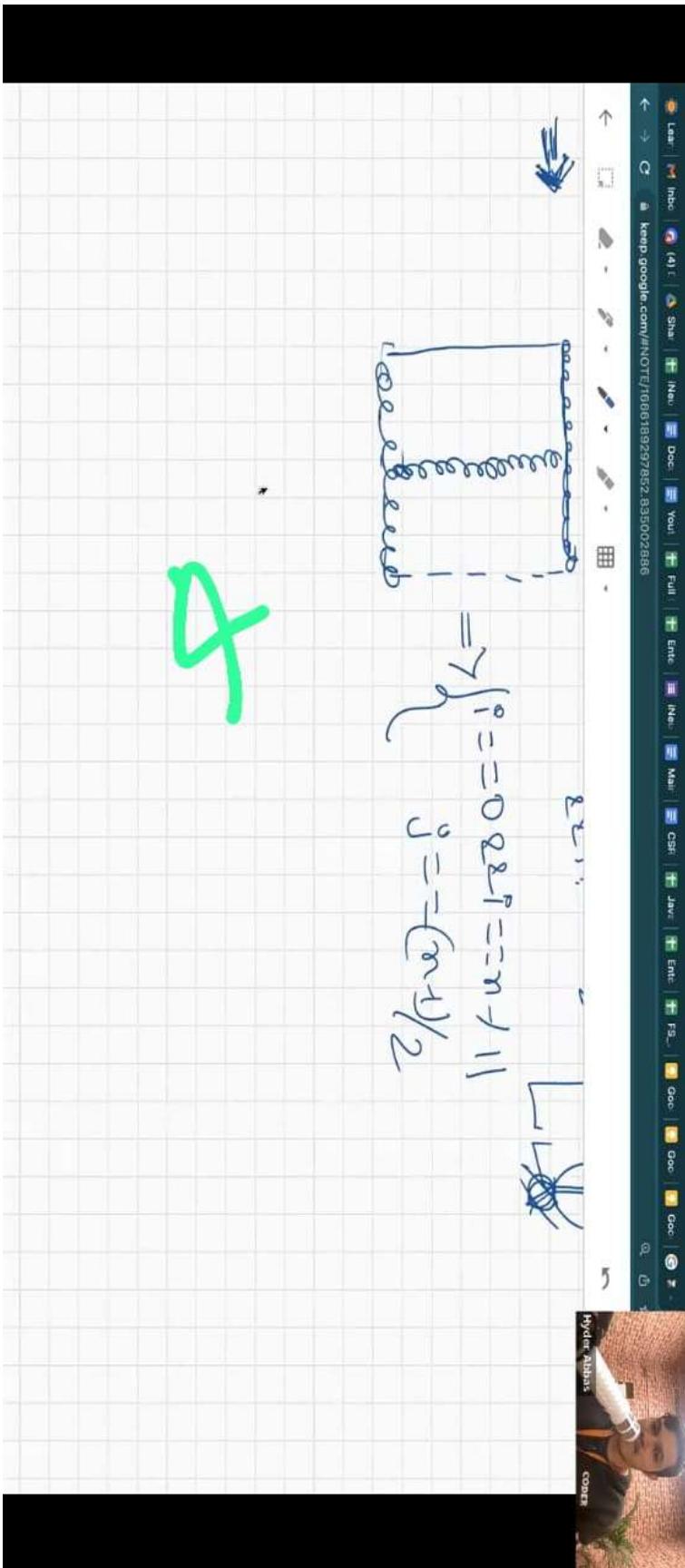
⇒ Q&A } ⇒ Questions

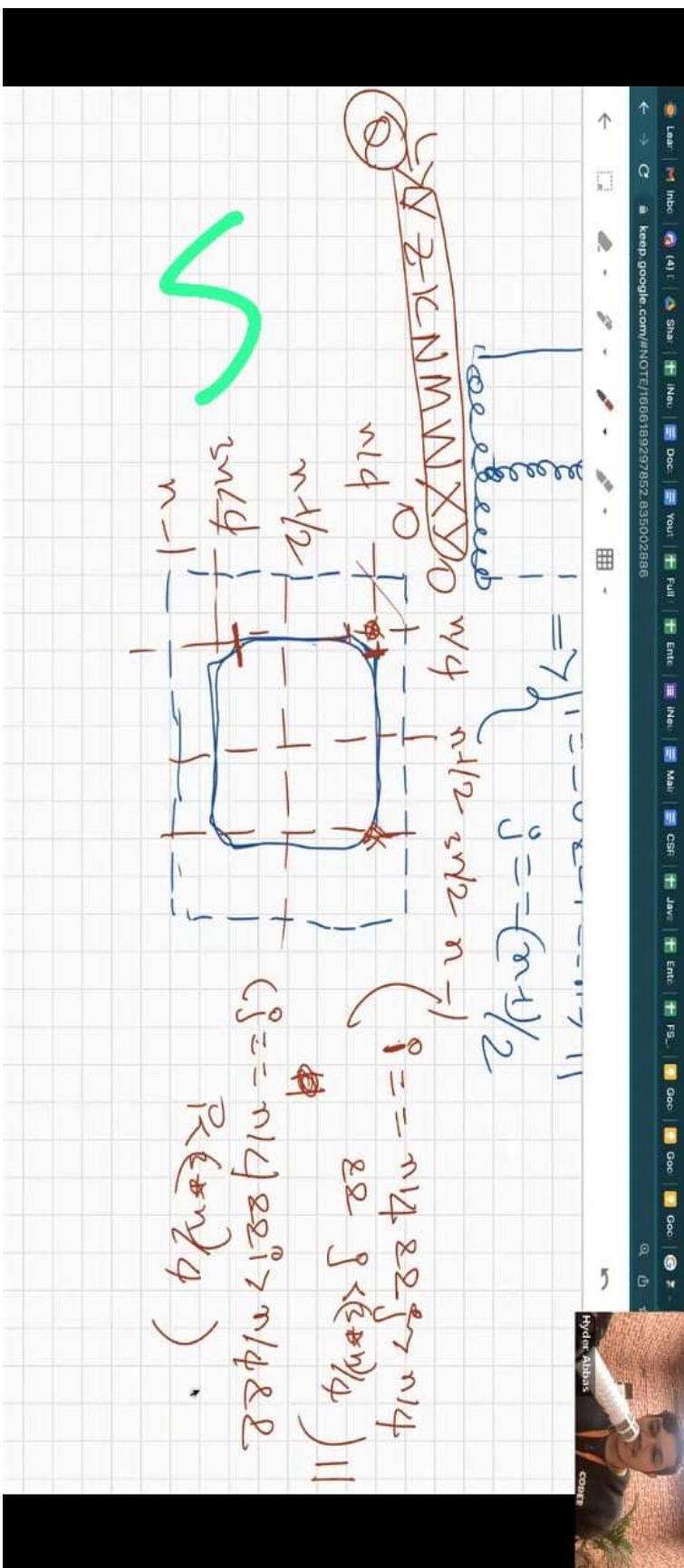
chat → knowledge / hrs --- class interaction notes











```
1 LaunchCpu   2 LaunchDawn  3 LaunchEve  4 LaunchGanymede  5 LaunchJupiter  6 LaunchLuna  7 LaunchMars  8 LaunchMercury  9 LaunchNeptune  10 LaunchPluto  11 LaunchSaturn  12 LaunchUranus  13 LaunchVesta  14 LaunchXenon  15 LaunchYmir  16 LaunchZembla  17 LaunchZeta  18 LaunchZeta  19 LaunchZeta  20 LaunchZeta  21 LaunchZeta  22 LaunchZeta  23 LaunchZeta  24 LaunchZeta  25 LaunchZeta  26 LaunchZeta  27 LaunchZeta  28 LaunchZeta  29 LaunchZeta
```

```
3
2 public class Launch4 {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int n=10;
7         for( int i=0; i<n; i++)
8             {
9                 for( int j=0; j<n; j++)
10                 {
11                     // G ( i==0 && j>0 && j< (3*n)/4 ) ||
12                     (j==0 && i>0 && i<n-1) ||
13                     (i==(n-1)/2 && j>=(n-1)/2 && j<=(3*n)/4) ||
14                     (j==(3*n)/4 && i>=(n-1)/2) ||
15                     (i==n-1 && j<(n-1)/2) ||
16                     (j==(n-1)/2 && i>=(n-1)/2) ||
17                     if()
18                 {
19                     System.out.print("*");
20                 }
21             else
22             {
23                 System.out.print(" ");
24             }
25         }
26     }
27 }
28
29 }
```

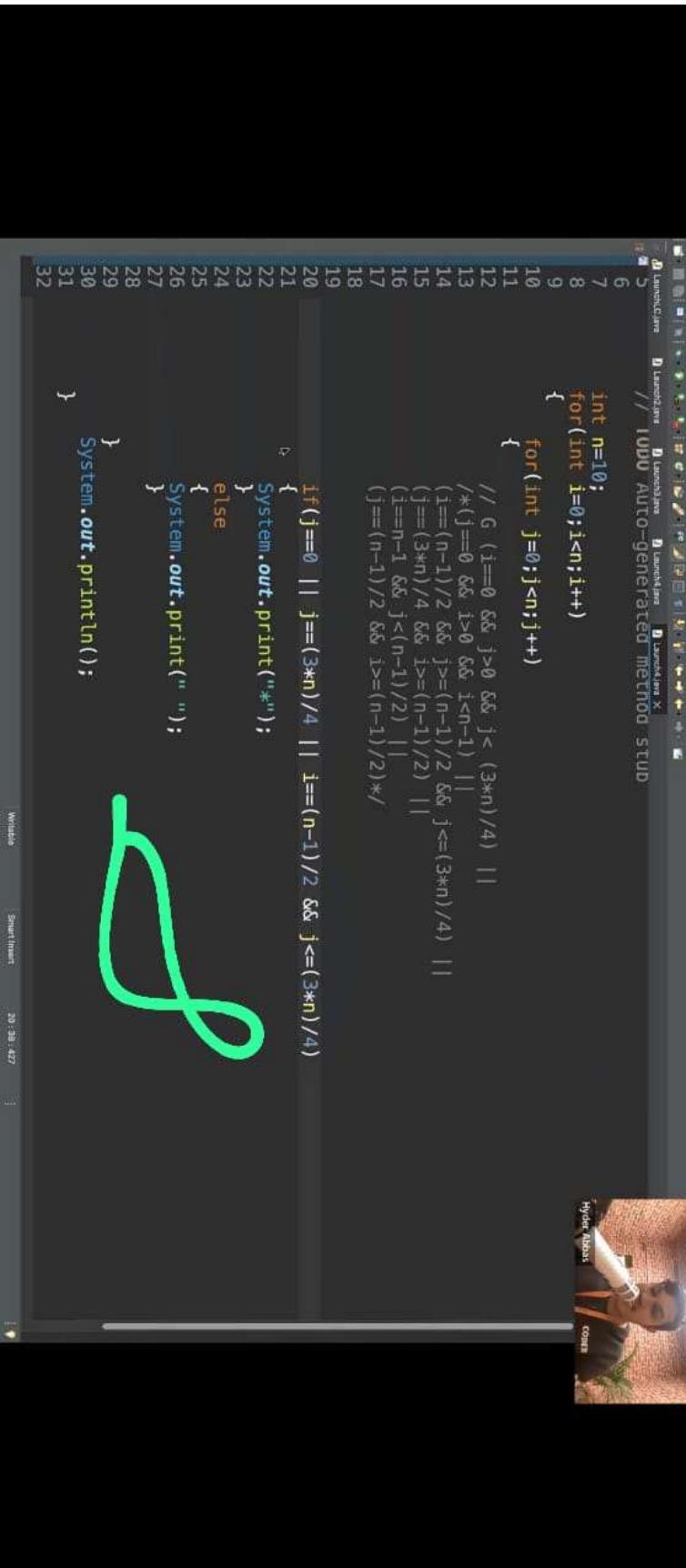
6

```
1  public class Launch4 {
2      public static void main(String[] args) {
3          // TODO Auto-generated method stub
4
5          int n=10;
6          for(int i=0;i<n;i++)
7          {
8              for( int j=0;j<n;j++)
9              {
10
11
12             if((i==0 && j<n-1) || j==0 ||(i==n-1 && j<n-1)|| (j==n-1 &&i>0 &&i<n-1) )
13             {
14                 System.out.print("*");
15             }
16         }
17     }
18
19     System.out.println(" ");
20 }
21
22 }
23 System.out.println();
24
25 }
26
27
28 .
29 }
```



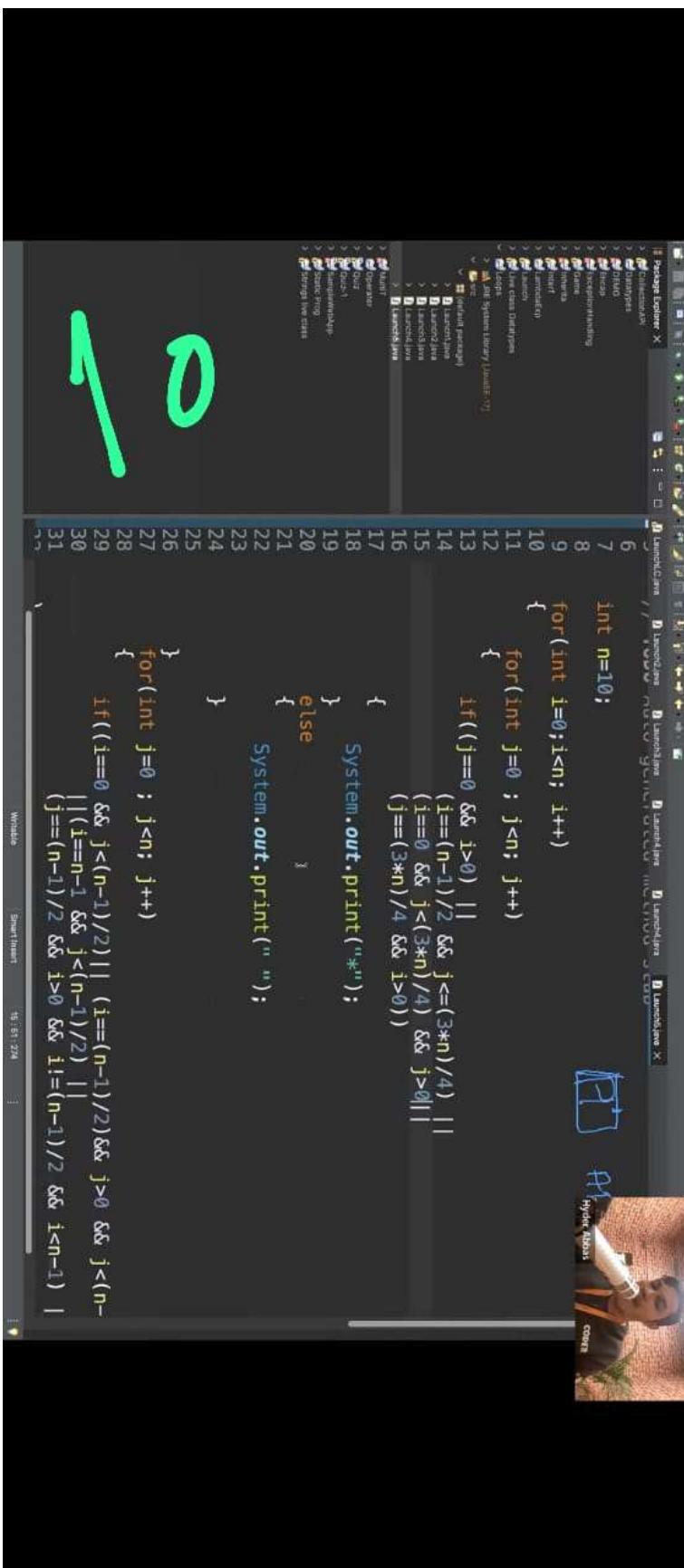
```
 6 // TODO AUTO-generated method stub
 7
 8 int n=10;
 9 for(int i=0;i<n;i++)
10 {
11     for(int j=0;j<n;j++)
12     {
13         // G ( i==0 && j>0 && j< (3*n)/4 ) ||
14         /*(j==0 && i>0 && i<n-1) ||
15         (i==(n-1)/2 && j>=(n-1)/2 && j<=(3*n)/4) ||
16         (j==(3*n)/4 && i>=(n-1)/2) ||
17         (i==n-1 && j<(n-1)/2) ||
18         (j==(n-1)/2 && i>=(n-1)/2)*/
19
20     if(j==0 || j==(3*n)/4 || i==(n-1)/2 && j<=(3*n)/4)
21     {
22         System.out.print("*");
23     }
24     else
25     {
26         System.out.print(" ");
27     }
28 }
29 System.out.println();
30
31 }
```

Q



```
10    for(int j=0;j<n;j++)
11    {
12        // G ( i==0 && j>0 && j< (3*n)/4 ) ||
13        /*(j==0 && i>0 && i<n-1) ||
14        (i==(n-1)/2 && j>=(n-1)/2 && j<=(3*n)/4) ||
15        (j==(3*n)/4 && i>=(n-1)/2) ||
16        (i==(n-1) && j<(n-1)/2) ||
17        (j==(n-1)/2 && i>=(n-1)/2)*/
18        //H j==0 || j==(3*n)/4 || i==(n-1)/2 && j<=(3*n)/4
19        if(j==0 || i==0 && j<(n-1)/2 || i==(n-1)/2 && j<(n-1)/2 ||
20        j==(n-1)/2 && i<(n-1)/2 && i>0)
21        {
22            System.out.print("*");
23        }
24        else
25        {
26            System.out.print(" ");
27        }
28    }
29
30    System.out.println();
31}
32}
33}
34}
35}
36}
37}
```

5



```
int n=10;
for(int i=0;i<n; i++)
{
    for(int j=0 ; j<n; j++)
    {
        if((j==0 && i>0) ||
           (i==(n-1)/2 && j<=(3*n)/4) ||
           (i==0 && j<(3*n)/4 && j>0) ||
           (j==(3*n)/4 && i>0))
        {
            System.out.print("*");
        }
        else
        {
            System.out.print(" ");
        }
    }
}
for(int j=0 ; j<n; j++)
{
    if((i==0 && j<(n-1)/2) ||
       (i==(n-1)/2 && j>0 && j<(n-1)/2) ||
       (j==(n-1)/2 && i>0 && i==(n-1)/2 && i<n-1) |
       (i==n-1 && j<(n-1)/2) ||
       (j==(n-1)/2 && i>0 && i==n-1) |
       (i==n-1 && j>0 && j<(n-1)/2) ||
       (j==(n-1)/2 && i>0 && i==n-1) )
    {
        System.out.print("*");
    }
    else
    {
        System.out.print(" ");
    }
}
```

```
int n=10;
for(int i=0;i<n; i++)
{
    for(int j=0 ; j<n; j++)
    {
        if((j==0 && i>0) ||
           (i==(n-1)/2 && j<(3*n)/4) ||
           ((i==0 && j<(3*n)/4) && j>0) ||
           (j==(3*n)/4 && i>0))
        {
            System.out.print("*");
        }
        else
        {
            System.out.print(" ");
        }
    }
}
for(int j=0 ; j<n; j++)
{
    if((i==0 && j<(n-1)/2) || (i==(n-1)/2)&& j>0 && j<(n-1)/2 ||(i==n-1 && j<(n-1)/2) ||(j==(n-1)/2 && i>0 && i==(n-1)/2 && i<n-1) |
```

```
21 package com.lambdasoup;
22
23 public class Diamond {
24     public static void main(String[] args) {
25         int n = 5;
26         for(int i=0 ; i<n; i++) {
27             for(int j=0 ; j<n; j++) {
28                 if((i==0 && j<(n-1)/2) || (i==(n-1)/2)&& j>0 && j<(n-1)/2
29                     ||(i==n-1 && j<(n-1)/2) ||
30                     (j==(n-1)/2 && i>0 && i!=(n-1)/2 && i<n-1) || j==0
31                 ) {
32                     System.out.print(" *");
33                 } else {
34                     System.out.print("   ");
35                 }
36             }
37             System.out.println();
38         }
39     }
40 }
41
42 }
43
44 }
45
46 {
47     System.out.print(" *");

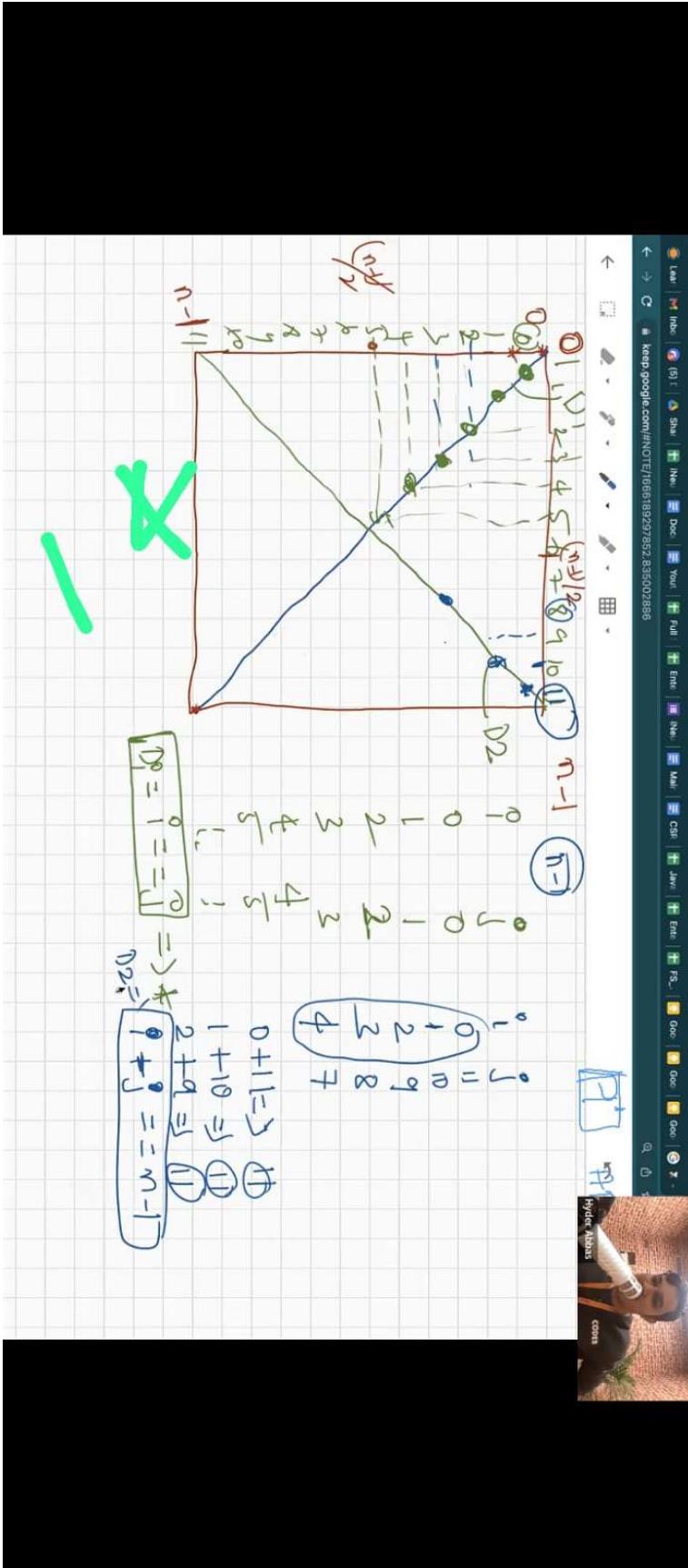
```

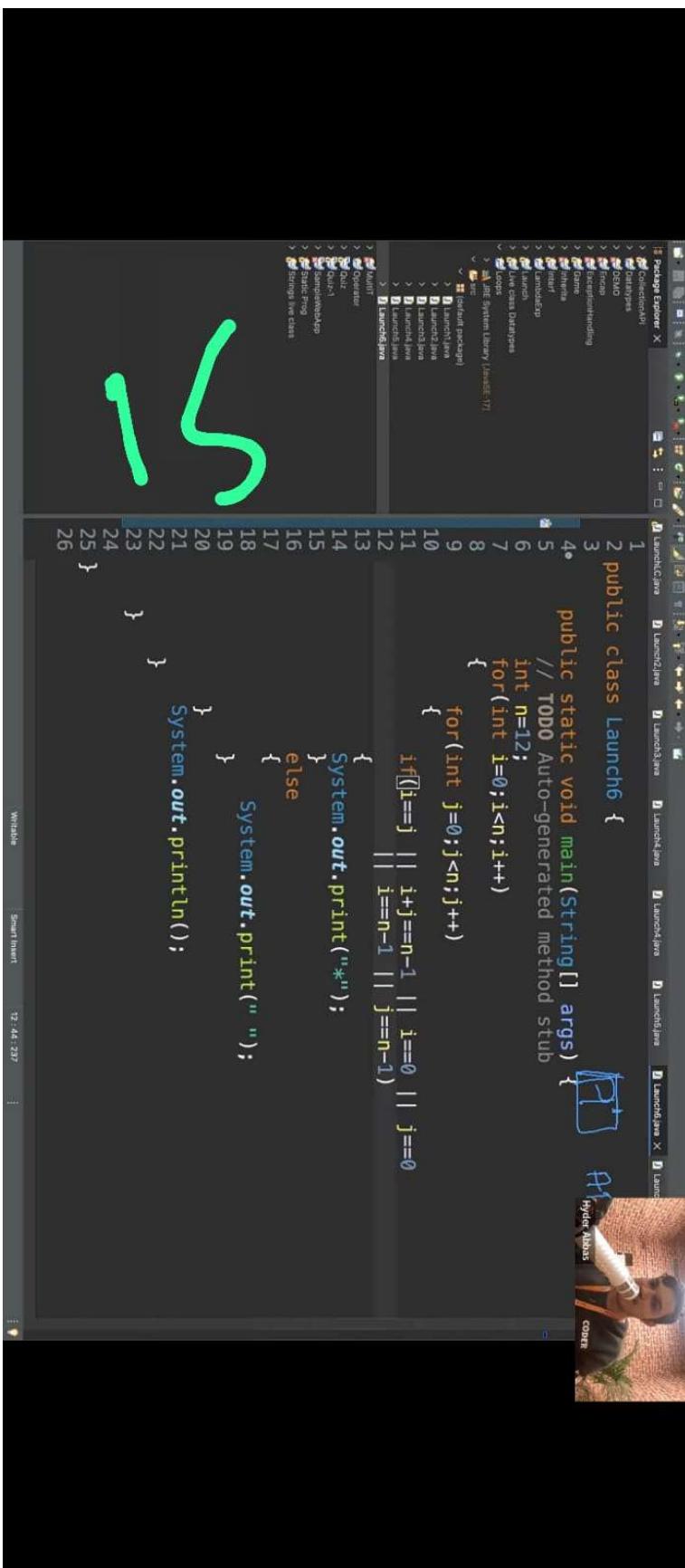


```
1  package com;
2
3  public class Main {
4      public static void main(String[] args) {
5          int n = 5;
6
7          for (int i = 0; i < n; i++) {
8              for (int j = 0; j < n; j++) {
9                  if (i == 0 && j > 0 && j < (3 * n) / 4 || i == n - 1 && j > 0 && j < (3 * n) / 4)
10                      System.out.print("*");
11                  else
12                      System.out.print(" ");
13              }
14              System.out.println();
15          }
16      }
17  }
```

The code is a diamond pattern printer. It uses nested loops to iterate through rows and columns. The pattern is defined by specific conditions for printing asterisks ('*') or spaces. The pattern is as follows:

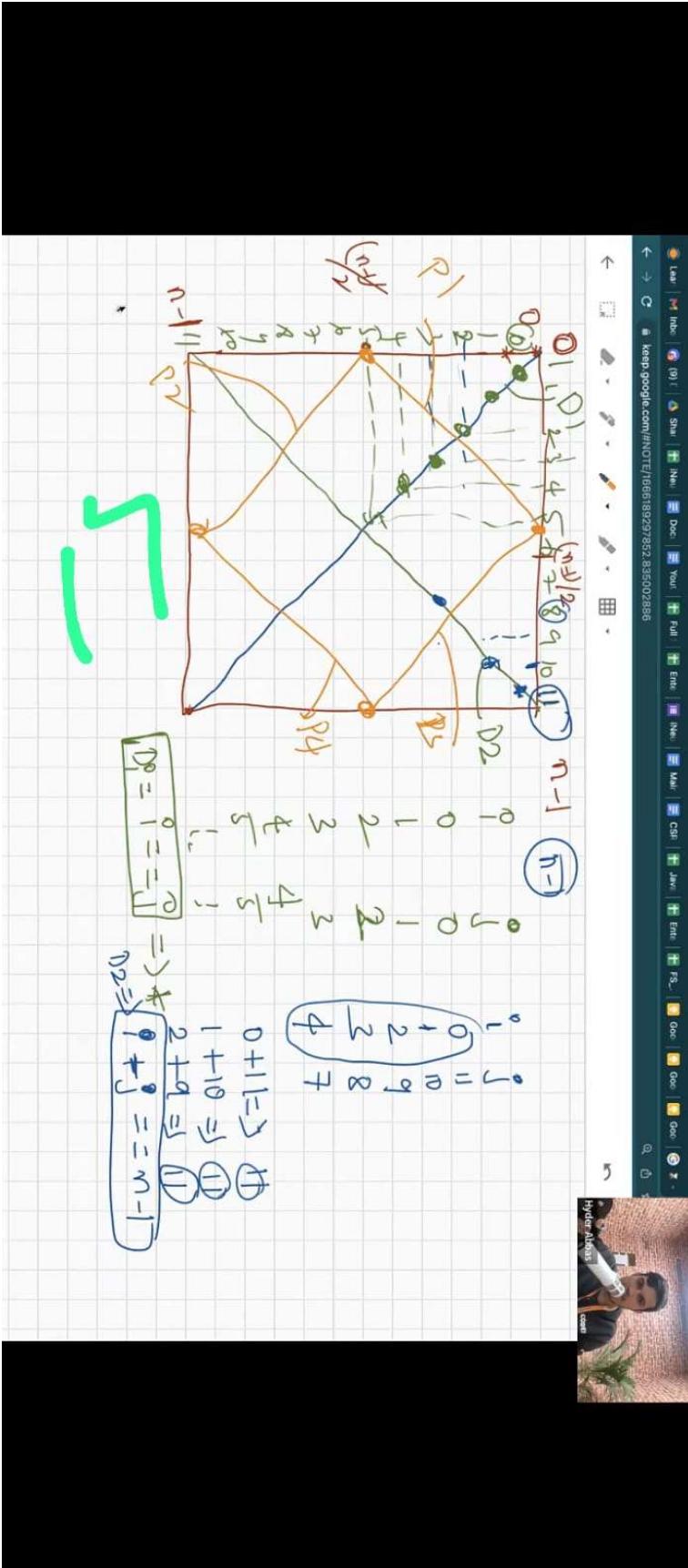
```
*****
 * * *
  * * *
   * * *
*****
```

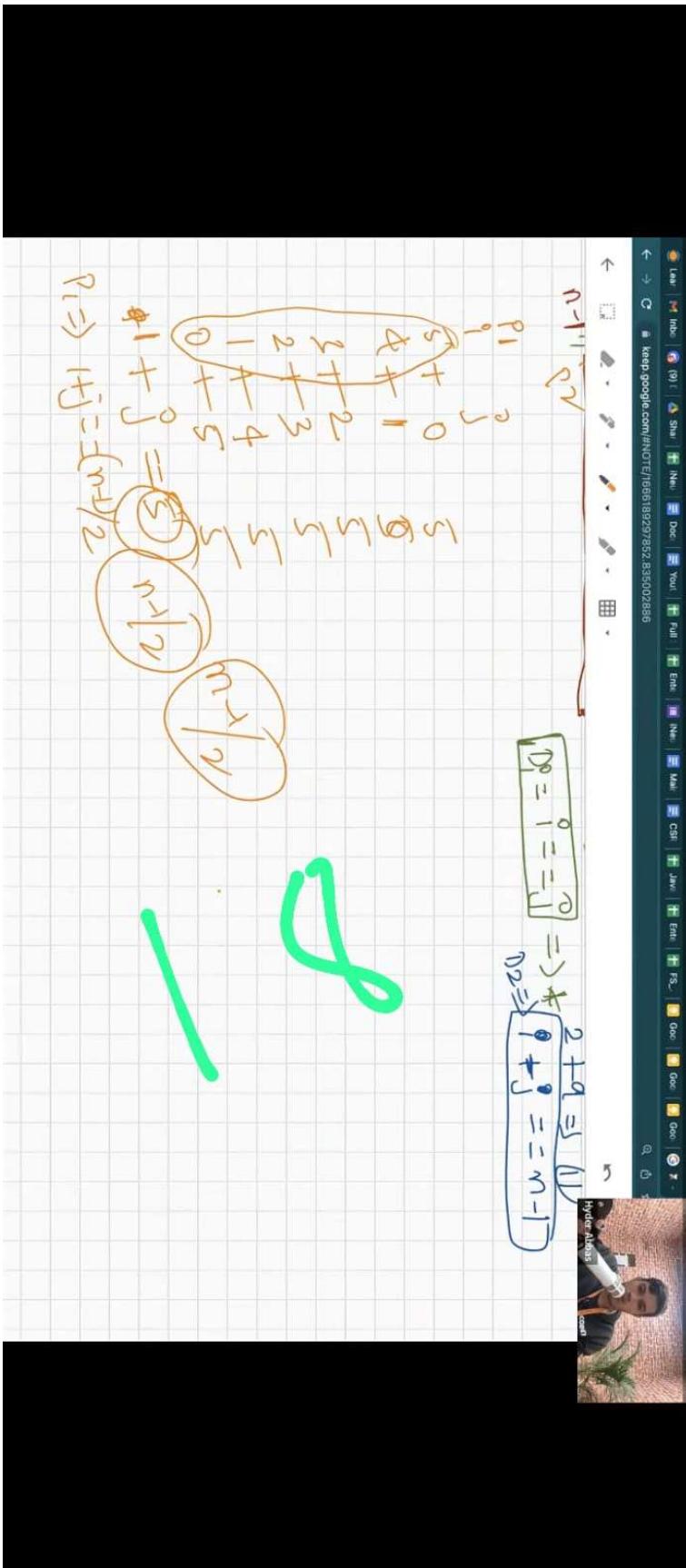


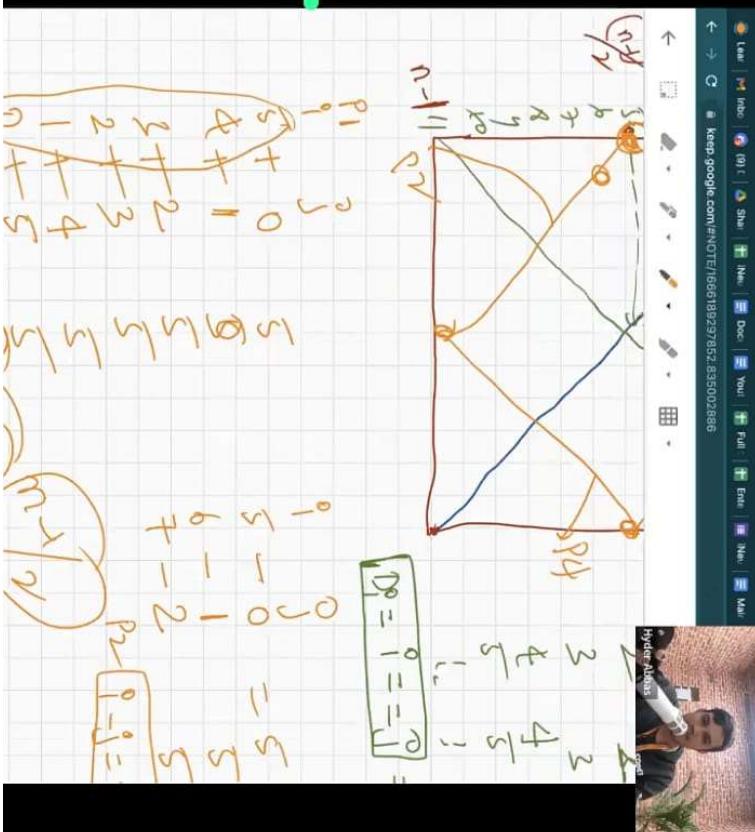
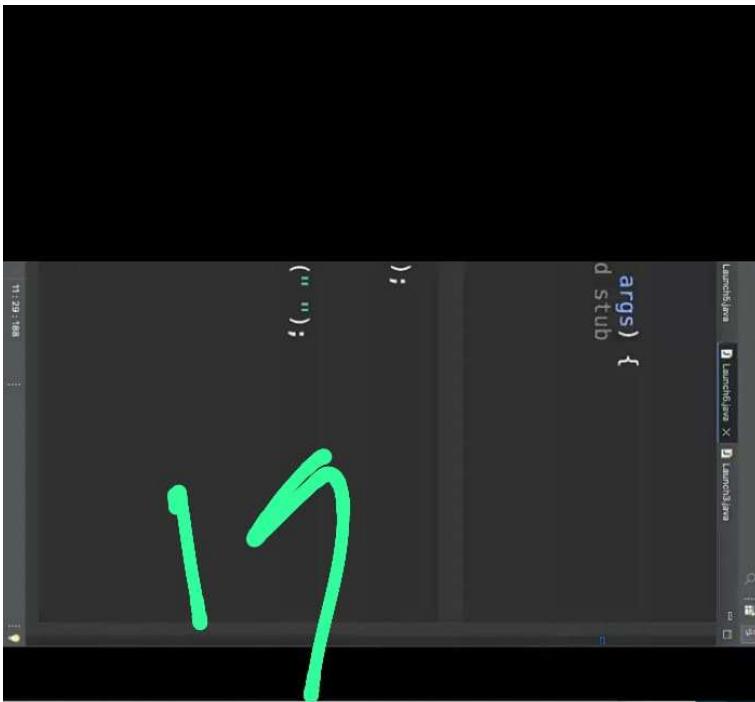


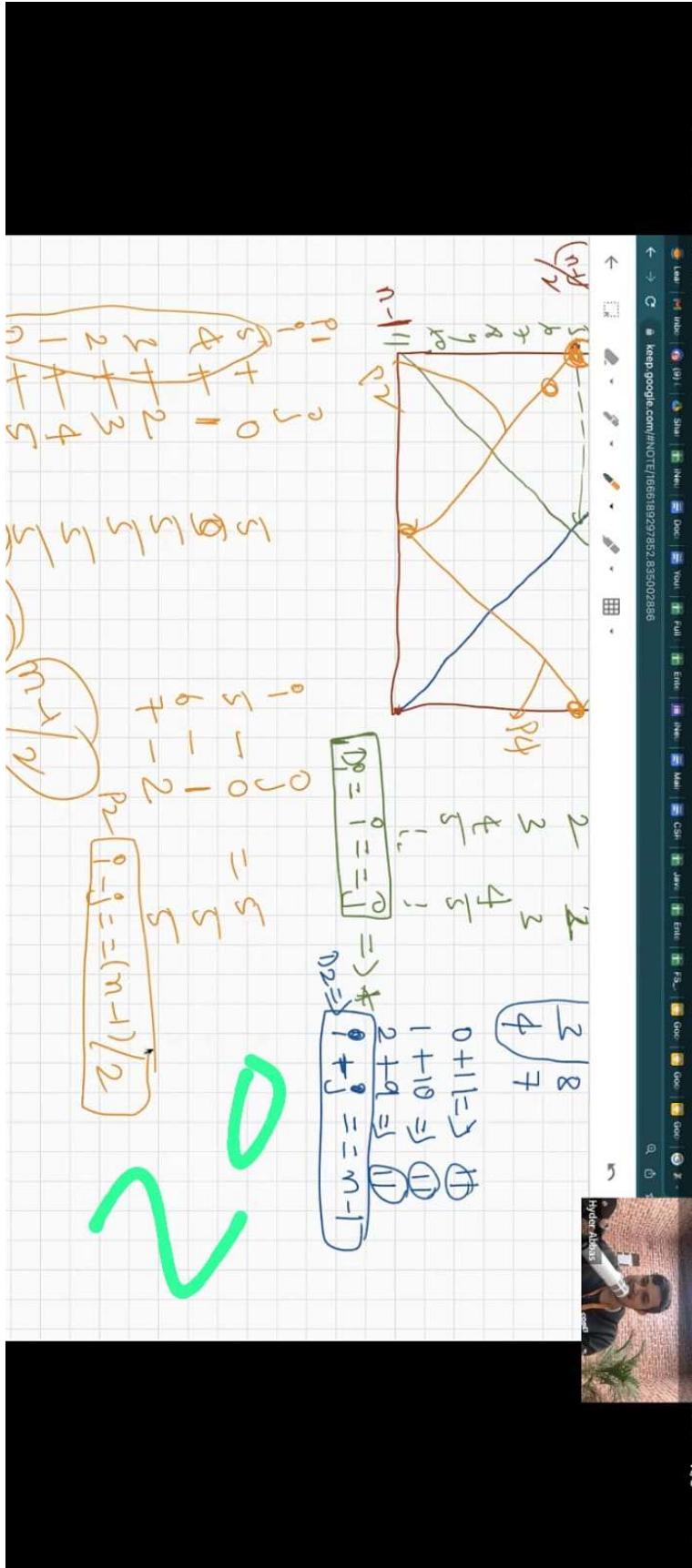
```
1 public class Launch6 {
2     // TODO Auto-generated method stub
3     int n=12;
4     for(int i=0;i<n;i++)
5     {
6         for(int j=0;j<n;j++)
7             if(i==j || i+j==n-1 || i==0 || j==0
8                 {
9                     System.out.print("*");
10                }
11            else
12            {
13                System.out.print(" ");
14            }
15        }
16    }
17
18    System.out.println();
19}
20}
21
22}
23}
24}
25}
26}
```

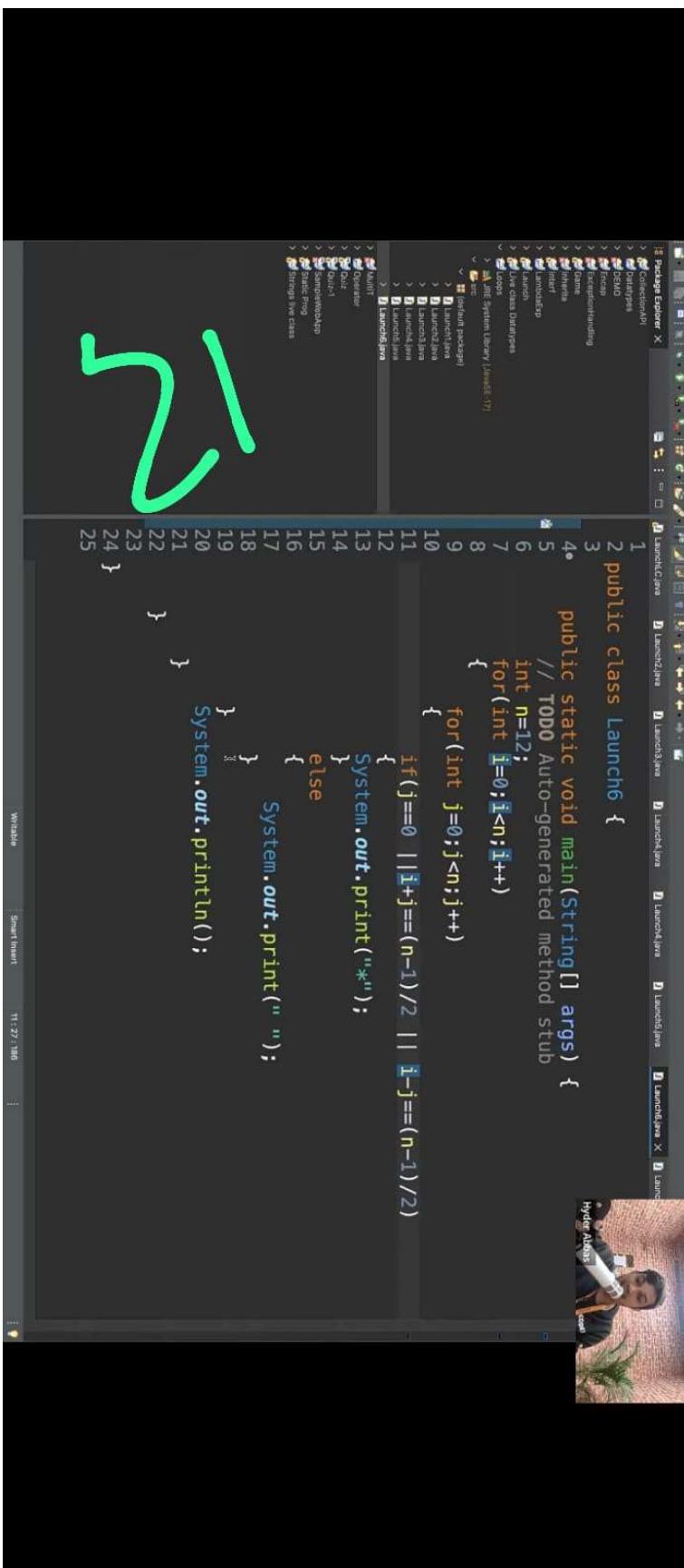
```
1 package Enginer;
2
3 public class Launch6 {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int n=12;
7         for(int i=0;i<n;i++) {
8             for(int j=0;j<n;j++) {
9                 if(i==j || j==0
10                     || j==n-1)
11                 {
12                     System.out.print("*");
13                 }
14             }
15         }
16     }
17     {
18         System.out.println();
19     }
20 }
21 }
22 }
23 }
24 }
25 }
26 }
```











A screenshot of a Java development environment. The code editor displays the following Java code:

```
1 package com.lambdaschool;
2
3 import java.util.Scanner;
4
5 public class Launch6 {
6     public static void main(String[] args) {
7         int n=12;
8         for(int i=0;i<n;i++) {
9             for(int j=0;j<n;j++) {
10                 if(j==0 || i+j==(n-1)/2 || i-j==(n-1)/2)
11                     System.out.print("*");
12                 else
13                     System.out.print(" ");
14             }
15         }
16     }
17 }
18
19
20
21
22
23
24
25 }
```

The code uses nested loops to print a diamond pattern of asterisks (*) and spaces. A green wavy underline highlights the opening brace of the inner loop at line 9. A video overlay of a person speaking is visible in the background of the IDE window.

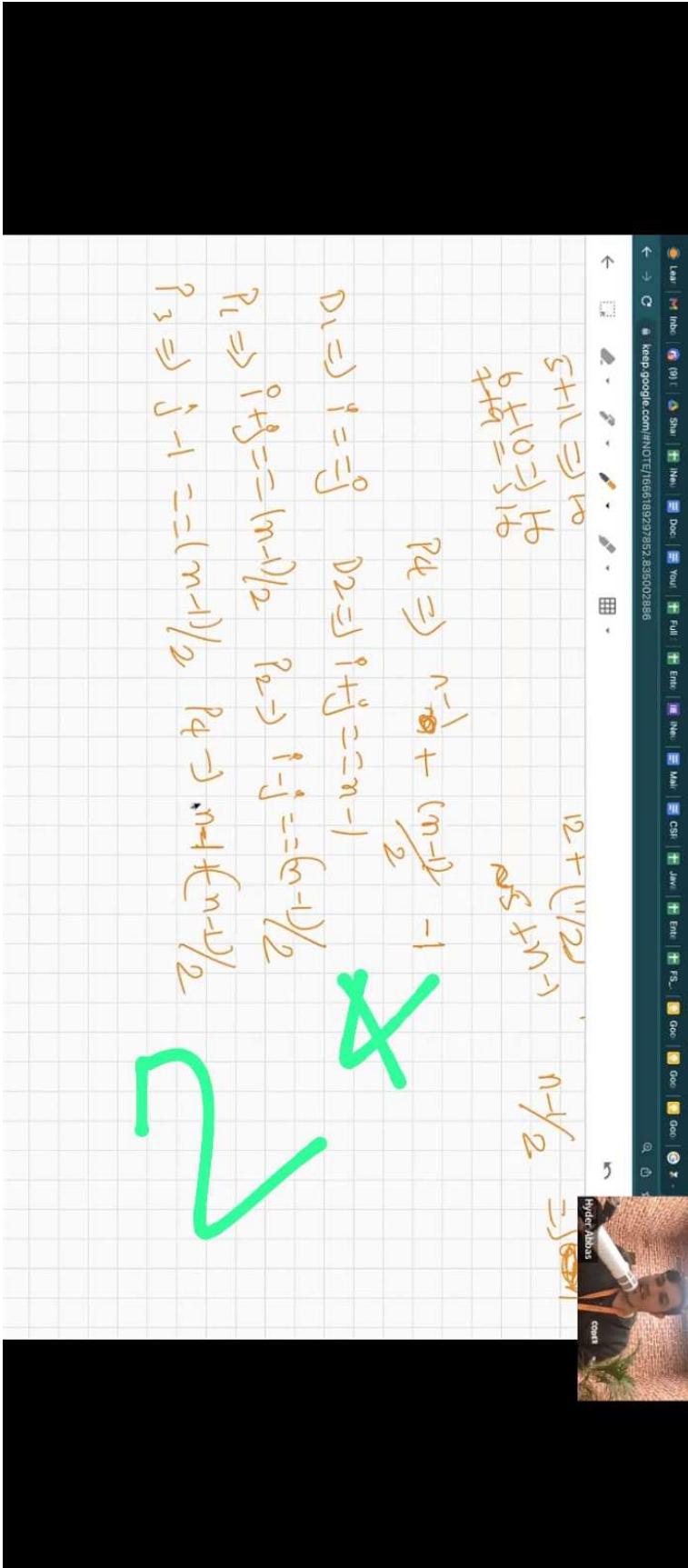
The image shows a Java development environment with the following details:

- Code Editor:** Displays the following Java code in a file named `Launch6.java`.

```
1 public class Launch6 {  
2     public static void main(String[] args) {  
3         // TODO Auto-generated method stub  
4         int n=13;  
5         for(int i=0;i<n;i++)  
6             for(int j=0;j<n;j++)  
7                 if((i+j==(n-1)/2 || i-j==(n-1)/2 ||  
8                     j-i==(n-1)/2 ||  
9                     i+j == n -1  
10                    + (n-1)/2))  
11                     System.out.print("*");  
12                 else  
13                     System.out.print(" ");  
14             }  
15         System.out.println();  
16     }  
17 }  
18 }  
19 }  
20 }  
21 }  
22 }  
23 }  
24 }  
25 }  
26 }  
27 }
```
- Package Explorer:** Shows a tree view of Java packages and classes, including `Launch6.java`, `Launch7.java`, `Launch8.java`, and `Launch9.java`.
- Terminal:** A small window at the bottom right shows the command `Hyder@Abbas`.

A screenshot of a Java IDE interface, likely Eclipse or IntelliJ IDEA, showing a code editor and a package explorer. The code editor displays a Java program with a large green hand-drawn mark over it.

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int n=28;
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(i+j==(n-1)/2 || i-j==(n-1)/2 || j-i == (n-1)/2 || i+j == n-1 || i==j || i+j == n-1
               || i==0 || j==0 || i==n-1 || j==n-1)
            {
                System.out.print("*");
            }
            else
            {
                System.out.print(" ");
            }
        }
        System.out.println();
    }
}
```



```
1 LaunchC.java
2 Launch2.java
3 Launch3.java
4 Launch4.java
5 Launch5.java
6 Launch6.java
7 Launch7.java
8 Launch8.java
9 Launch9.java
10 Launch10.java
11 Launch11.java
12 Launch12.java
13 Launch13.java
14 Launch14.java
15 Launch15.java
16 Launch16.java
17 Launch17.java
18 Launch18.java
19 Launch19.java
20 Launch20.java
21 Launch21.java
22 Launch22.java
23 Launch23.java
24 Launch24.java
25 Launch25.java
26 Launch26.java
27
```

2
Z

```
2b
12 Package Explorer X
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

3 4.0c static void main(String[] args) {
5 / TODO Auto-generated method stub
6
7     int n=11;
8     for(int i=0;i<n;i++)
9     {
10         for(int j=0;j<n;j++)
11         {
12             if( i+j>=(n-1)/2 && j<=(n-1)/2 && i<=(n-1)/2 ||
13                 i-j<=(n-1)/2 && i>=(n-1)/2 && j<=(n-1)/2 ||
14                 j-i<=(n-1)/2 && j>=(n-1)/2 && i<=(n-1)/2 ||
15                 i+j <= n-1 + (n-1)/2 && i>(n-1)/2 && j>(n-1)/2 )
16             )
17         }
18         {
19             System.out.print("*");
20         }
21     else
22     {
23         System.out.print(" ");
24     }
25     System.out.println();
26
27
28
29
```


19-10-2022_snippets-classmates - Notepad

File Edit View

Q>
Give
int i=10;//10
int j=20;//30
int k=(j+i)/5;//
k = (j+i)/5
k = (j=20+10)/5
k = (j=30)/5
k = 30/5
k= 6
System.out.println([i+"," +j +"," +k);

W8

A.10:30:6
B.10:22:22
C.10:22:20
D.10:22:6

Answer : A

22°C
Rain 0.100



Ln 63 Col 11

100% Windows (C:\UF7) UTF-8

ENG IN 19-10-2022

```
if(x){//CE: unexpected type required: boolean,found:int
    System.out.println("hello");
}else{
    System.out.println("hiee");
}
A. hello
B. hiee
C. CompileTime Error
D. Some problem by jvm at the execution
E. None of the above
```

- C. CompileTime Error
- D. Some problem by jvm at the execution
- E. None of the above

2A

Answer: C



File Edit View

E. None of the above

Answer: C

```
Q>
int x=10;
if(x>20){//E: unexpected type required: boolean,found:int
    System.out.println("hello");
}
else{
    System.out.println("hiee");
}
A. hello
B. hiee
C. CompileTime Error
D. Some problem by jvm at the execution
E. None of the above
```

Answer: C

Lh 97 Col 59

100% Windows (CRLF) UTF-8

19-10-2022 snippets classmate - Notepad

File Edit View

Q>

```
int x=10;
if(x==20)//operator used is Equality operator ==, != output is boolean
System.out.println("Hello");
else{
System.out.println("hiee");
}
A. hello
B. hiee
C. CompileTime Error
D. Some problem by jvm at the execution
E. None of the above
```

Answer: B

100% Windows (CRLF) UTF-8

~ ⌂ ⌄ ENG ⌄ 22:30 19-10-2022

22°C Rain to stop

Nitin Manjrekar



Q>

```
boolean b=false;
if(b=true){//assignment operator is evaluated on boolean type, JVM if(true)
    System.out.println("Hello");
} else{
    System.out.println("hiee");
}
```

A. hello
B. hiee
C. CompileTime Error
D. Some problem by jvm at the execution
E. None of the above

Answer : A



19-10-2022_snippets-classnotes - Notepad

File Edit View

Q>

```
boolean b=false;
if(b==true){//Equality operator result is boolean type , JVM if(false == true)---->if(false)
    System.out.println("hello");
} else{
    System.out.println("hiee");
}
```

- A. hello
- B. hiee
- C. CompileTime Error
- D. Some problem by jvm at the execution
- E. None of the above

Answer: B



Ln 154, Col 1
22°C Satisfactory air

19-10-2022

22:44

100% Windows (C|E)

UTF-8

19-10-2022

22:44



*19-10-2022_snippets.classnames - Notepad

File Edit View

Answer: B

Q>

```
if(boolean)
stmt-1;
if(true)
System.out.println("hello");
```

Note: if there is only statement which needs to be a part of if, then {} is optional.

A. Compile Time Error
B. hello
C. Some problem by jvm at the execution
D. None of the above

Answer : B



File Edit View

Answer : D(becoz , is also valid java statement)

Q>

Note: if there is only statement which needs to be a part of if, then {} is optional, but that statement should not be a declarative statement.

```
public class Test{  
    public static void main(String args[]){  
        if(true)  
            int x=10; //CE: declaration not allowed here  
    }  
}
```

A. Compile Time Error
B. hello
C. Some problem by jvm at the execution
D. No Output

Answer: A

Q|

199 Col 2 100% Windows (CRLF) UTF-8 22:40 19-10-2022

22°C Rain to stop

19-10-2022_snippets-classnotes - Notepad

File Edit View

Answer: A

Q>

```
public class Test{  
    public static void main(String args[]){  
        if(true){  
            int x=10; //valid for compiler becoz of {}  
        }  
    }  
}
```

A. Compile Time Error
B. hello
C. Some problem by jvm at the execution
D. No Output

Answer: D

public class Test{
 public static void main(String args[]){
 if(true)
 System.out.println("Hello");
 }
}

19-10-2022 22:41
22°C Rain to stop

Q E

Nitin Manjrekar

File Edit View

Answer: D

Q>

```
public class Test{  
    public static void main(String args[]){  
        if(true)  
            System.out.println("Hello");//Independent of if statement  
        System.out.println("Hi"); //Independent of if statement  
    }  
}
```

How many statements are independent of if?

- A. 0-stmt
- B. 1-stmt
- C. 2-stmt
- D. 3-stmt

Answer: B



Ln 230, Col 1

22°C Rain to stop



100%

Windows (C:\ELF)

UTF-8

~ ⌂ ⌄ ENG ⌄ 22:43

19.10.2022

