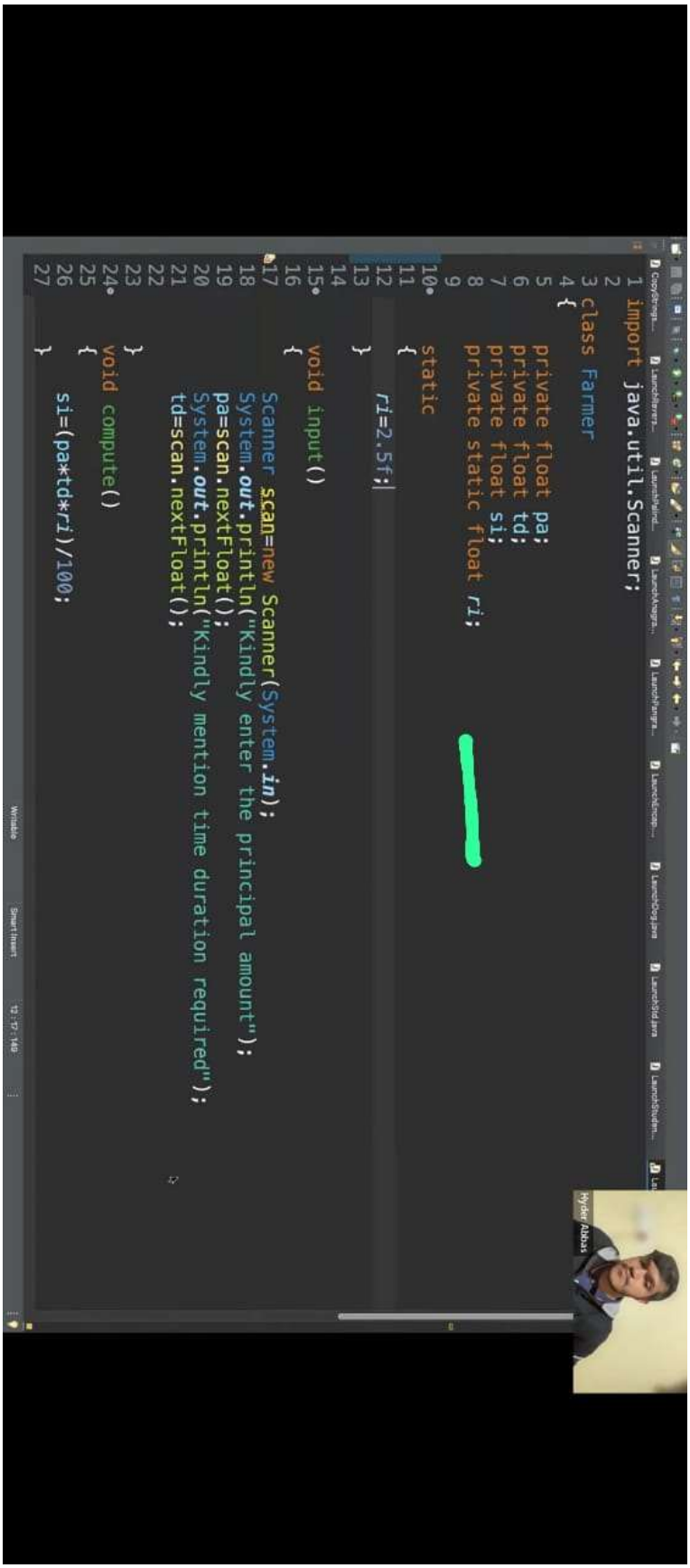
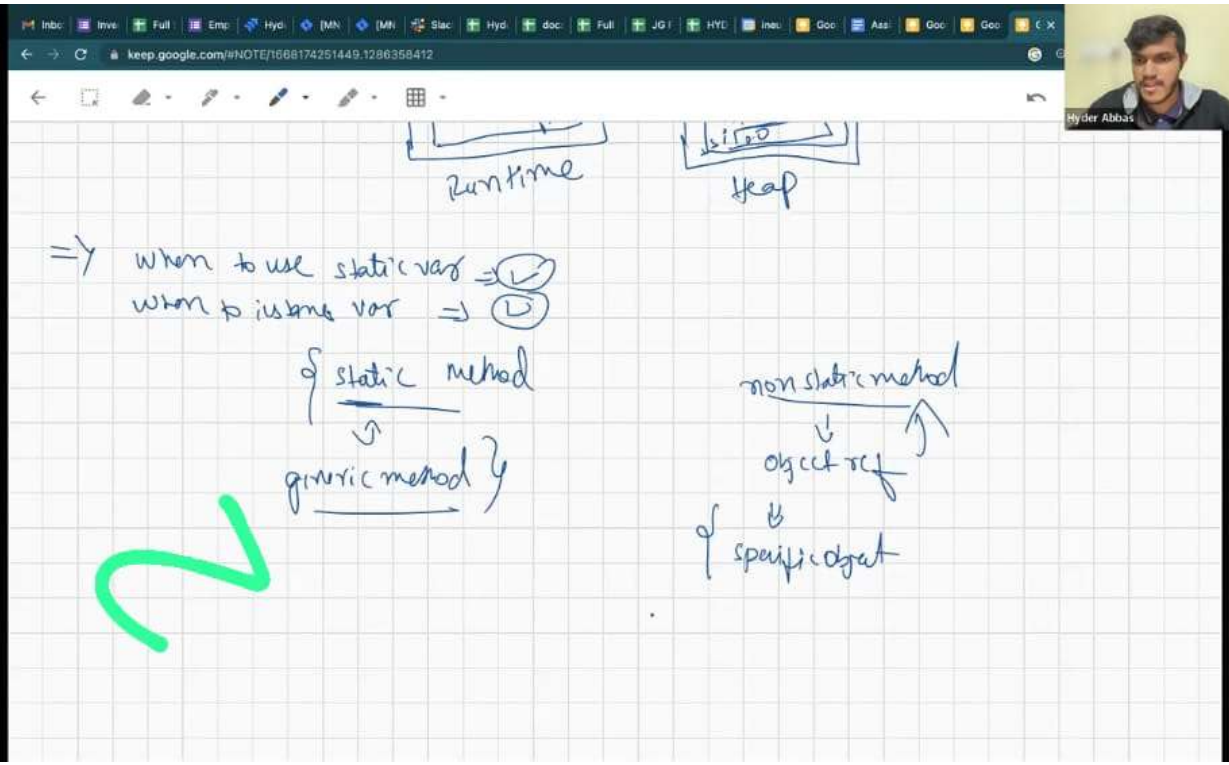
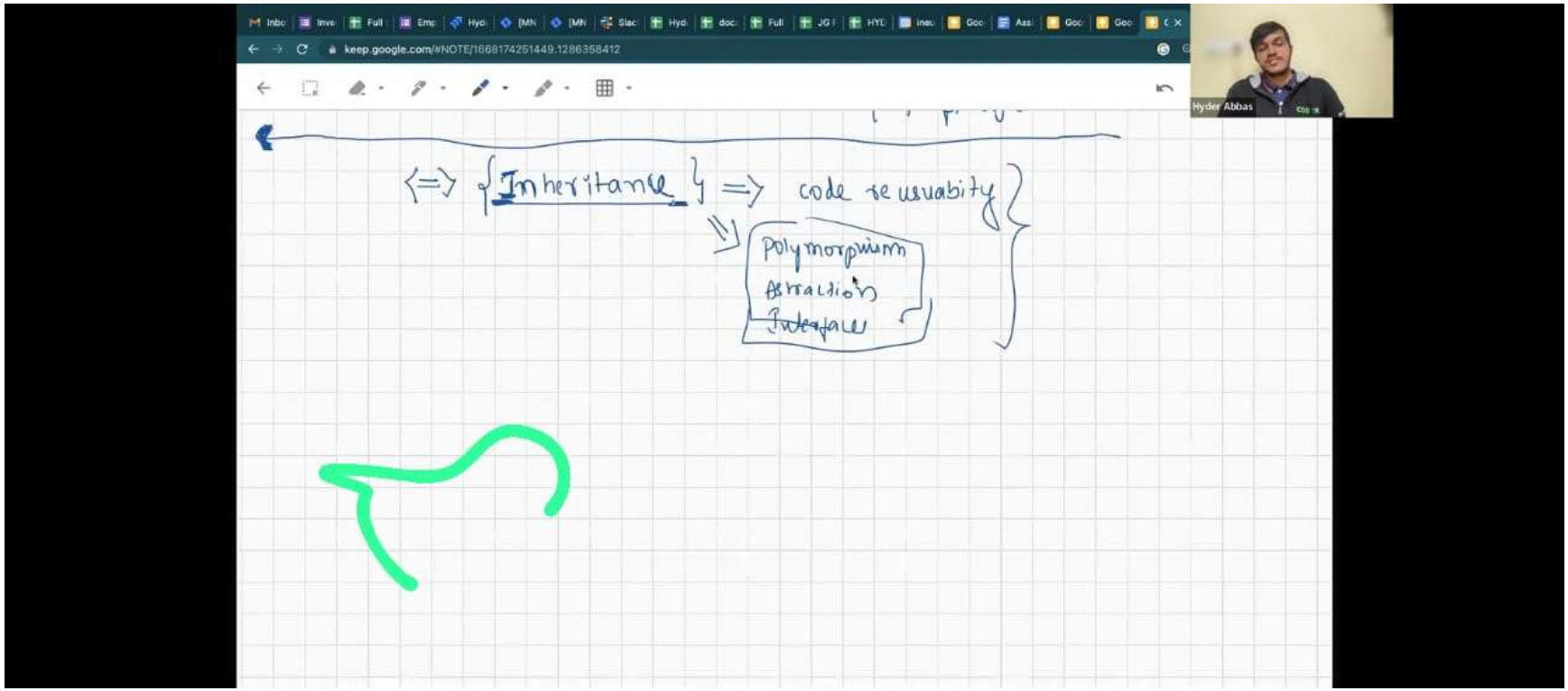
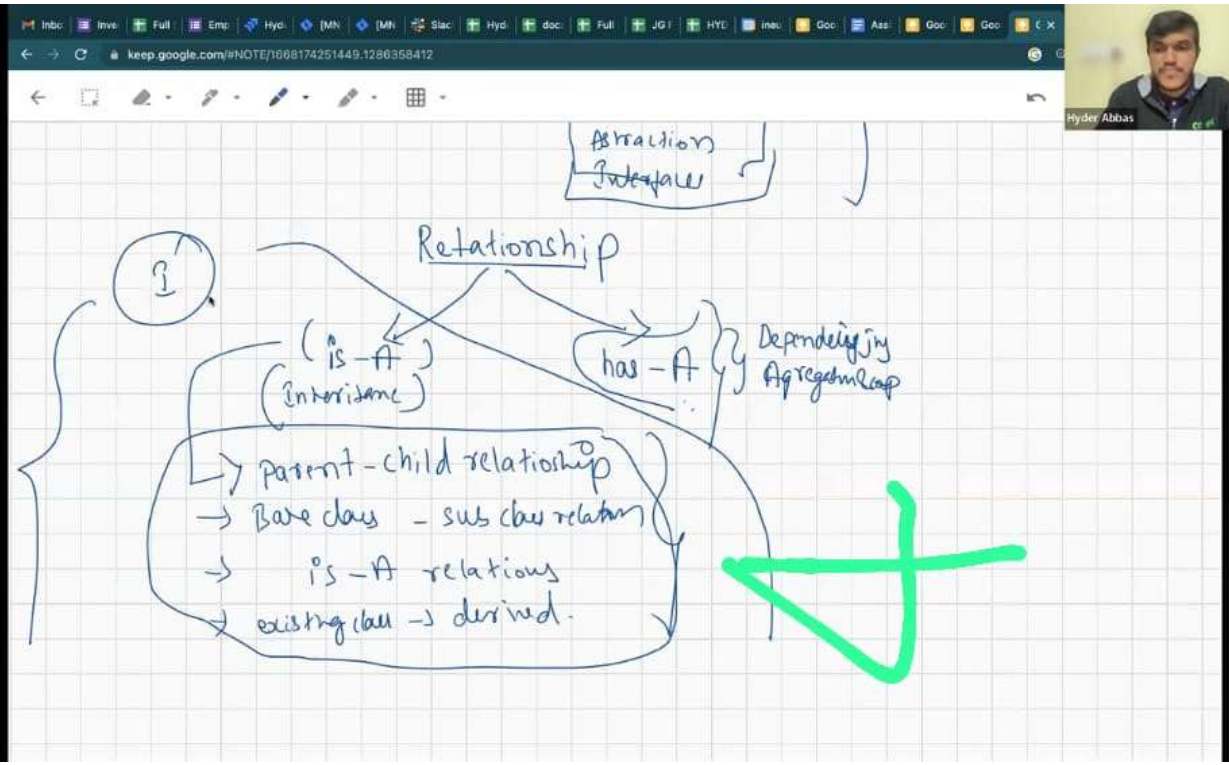
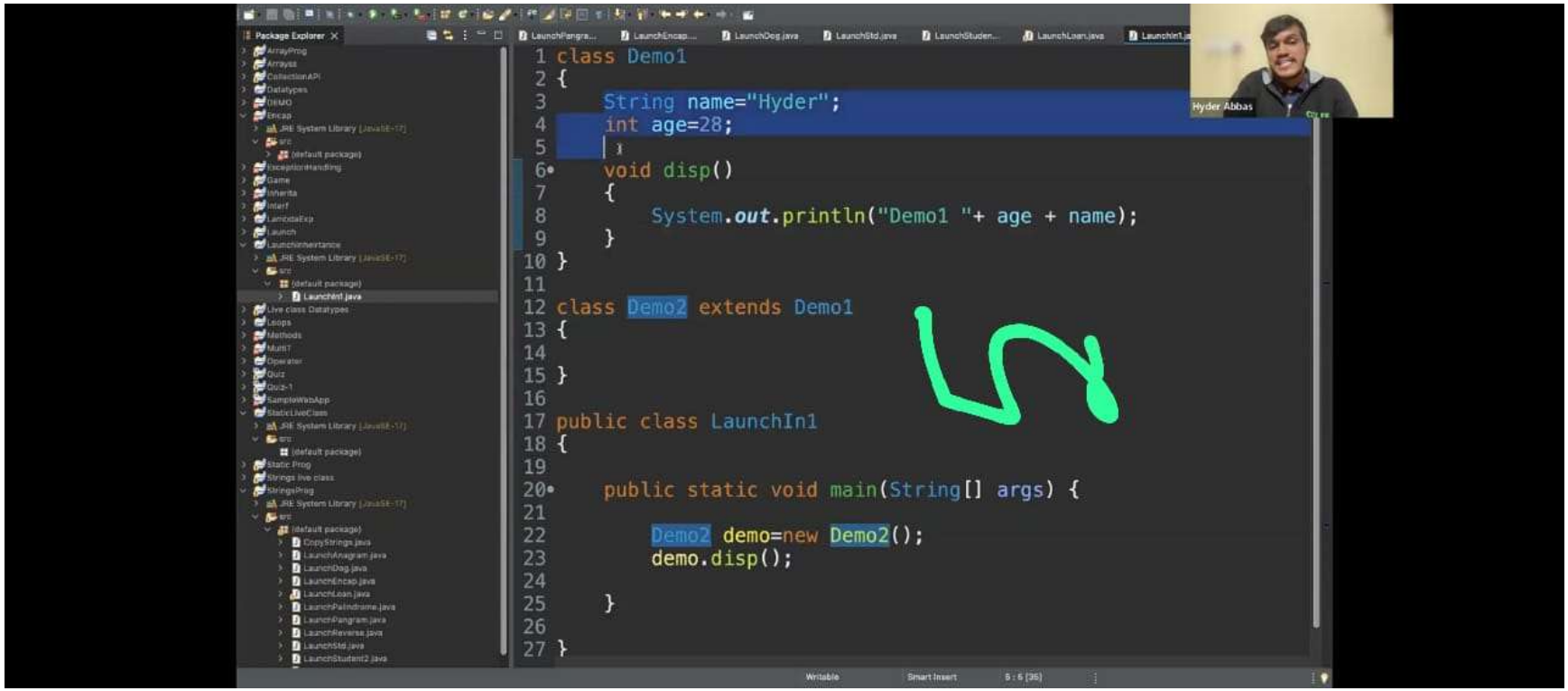
Static in Java

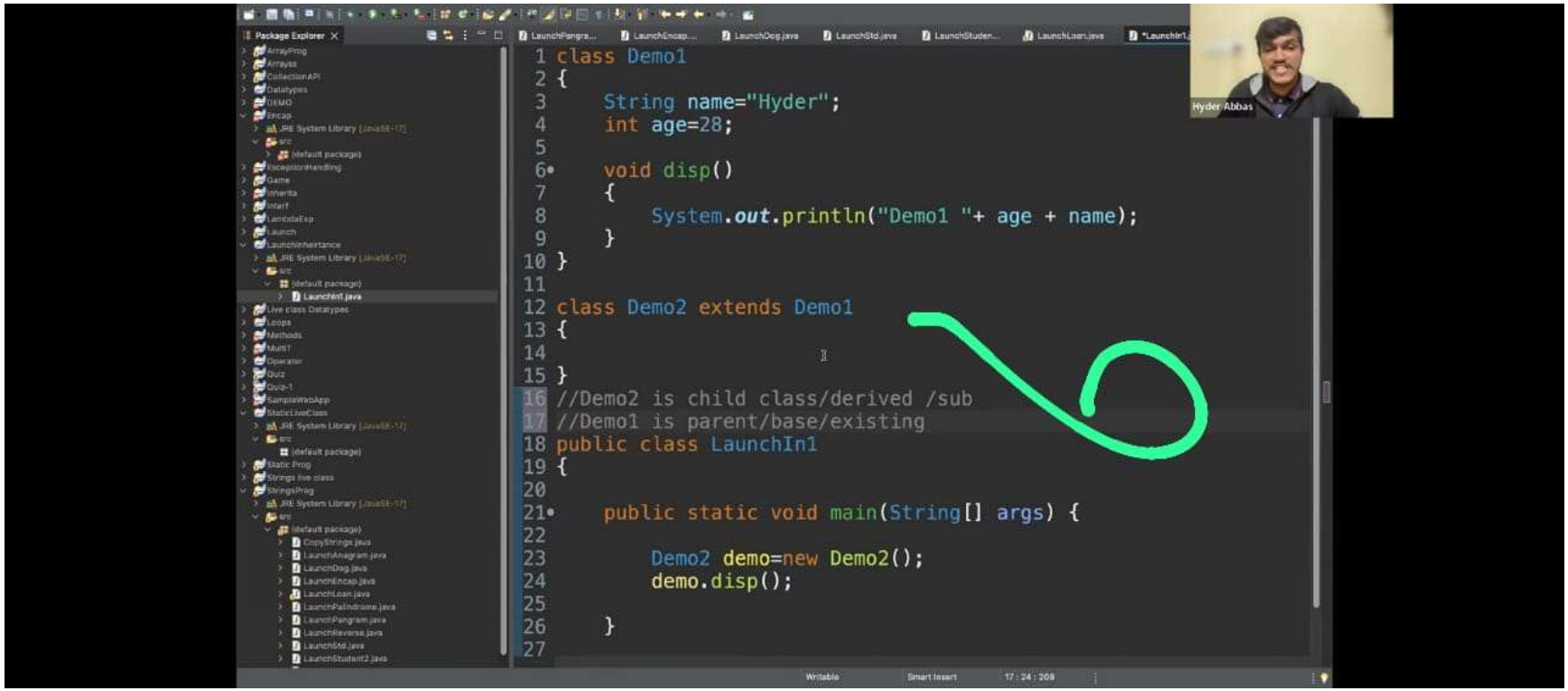


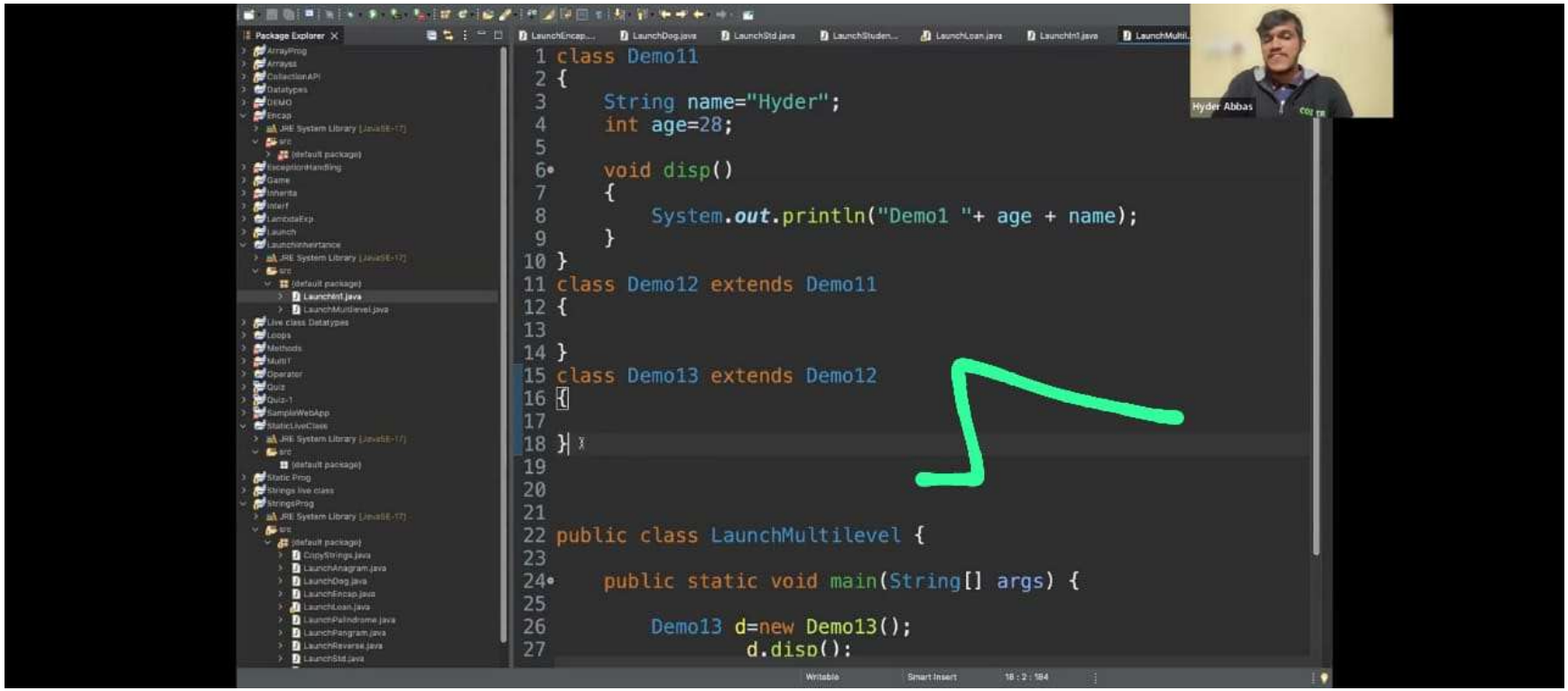

keep.google.com/#NOTE/1668174251449.1286358412
Runtime
Heap
=> when to use static var => ✓
when to use instance var => ✗
static method
generic method
non static method
object ref
specific object
Hyder Abbas

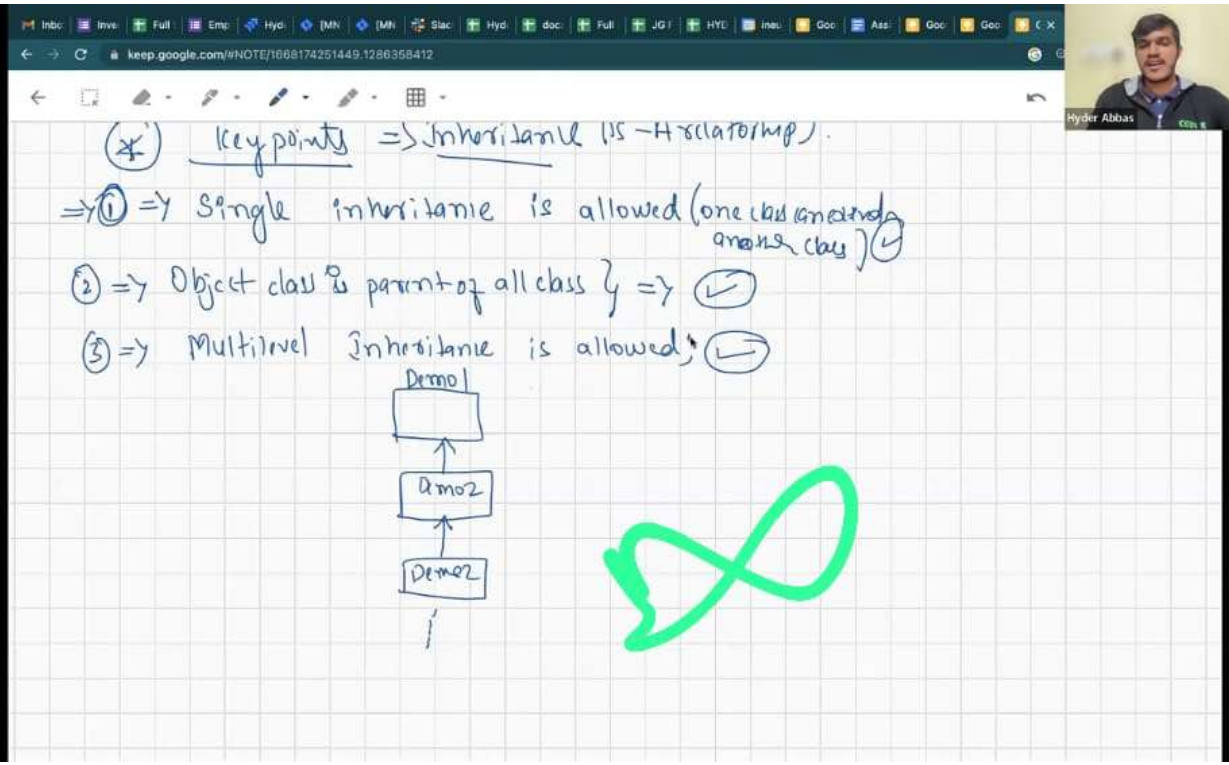

keep.google.com/#NOTE/1668174251449.1286358412
Hyder Abbas
<=> {Inheritance} => code reusability
Polymorphism
Abstraction
Interfaces

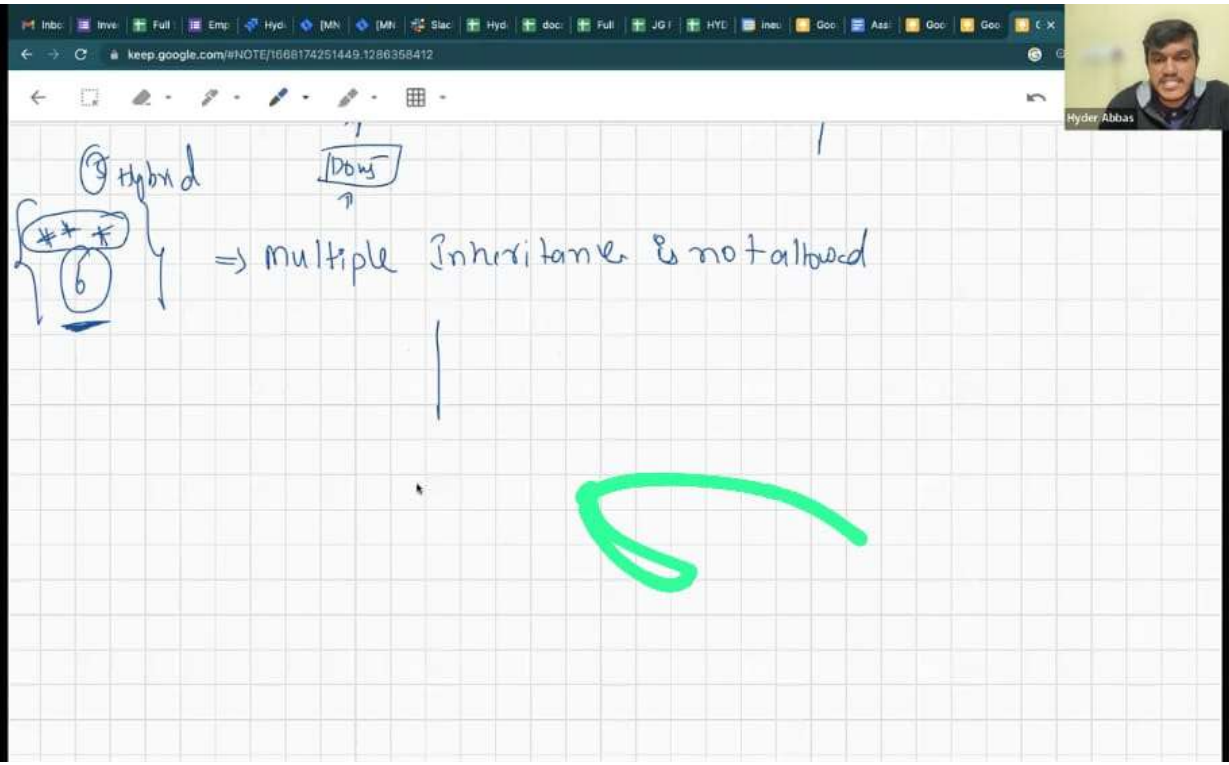

keep.google.com/#NOTE/1668174251449.1286358412
Abstraction
Interface
Relationship
I
(is-A)
(Inheritance)
has-A
Depending on Aggregation
Parent-child relationship
→ Base class - sub class relation
→ is-A relations
→ existing class → derived.
Hyder Abbas

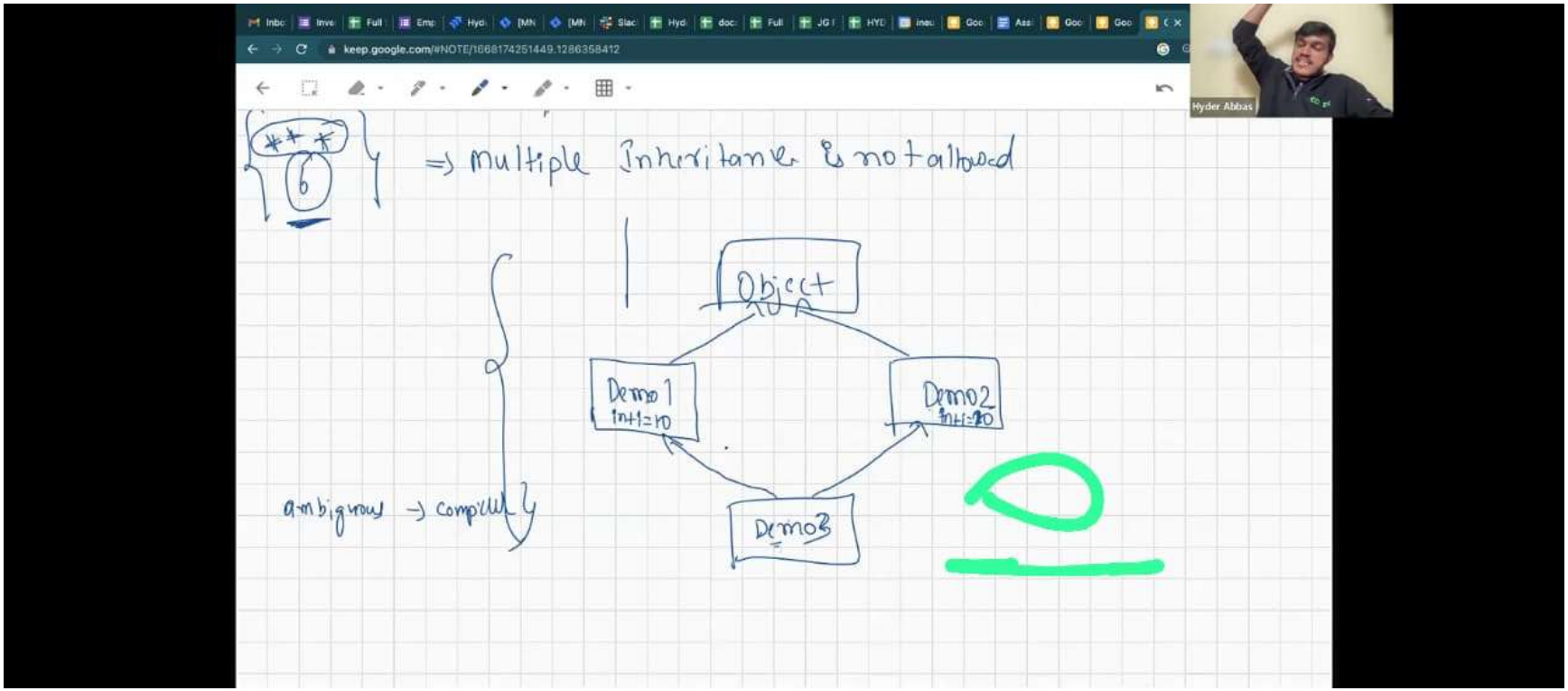

Package Explorer
class Demo1
{
String name="Hyder";
int age=28;
}
void disp()
{
System.out.println("Demo1 "+ age + name);
}
}
class Demo2 extends Demo1
{
}
public class LaunchIn1
{
public static void main(String[] args) {
Demo2 demo=new Demo2();
demo.disp();
}
}
Hyder Abbas

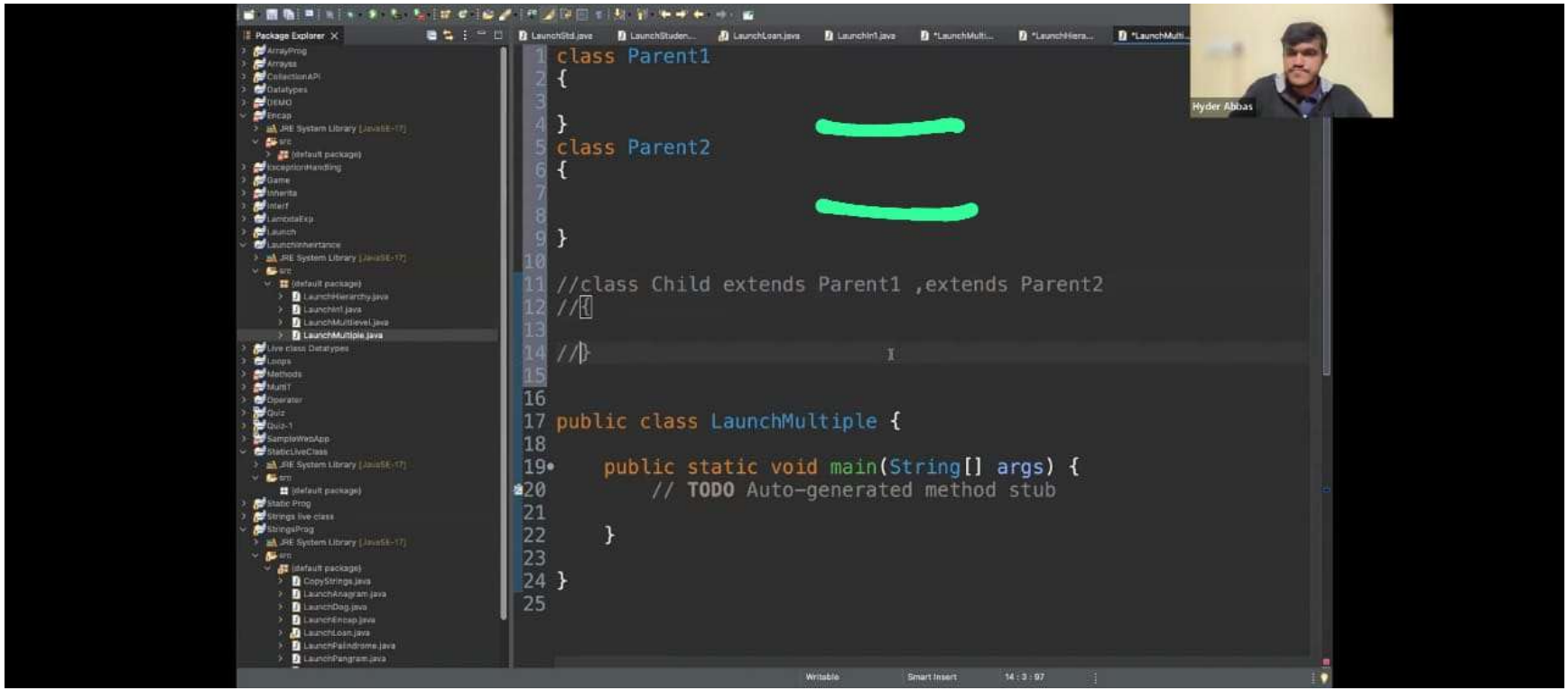

Package Explorer
LaunchIn1.java
1 class Demo1
2 {
3 String name="Hyder";
4 int age=28;
5
6 void disp()
7 {
8 System.out.println("Demo1 "+ age + name);
9 }
10 }
11
12 class Demo2 extends Demo1
13 {
14
15 }
16 //Demo2 is child class/derived /sub
17 //Demo1 is parent/base/existing
18 public class LaunchIn1
19 {
20
21 public static void main(String[] args) {
22
23 Demo2 demo=new Demo2();
24 demo.disp();
25
26 }
27
Hyder Abbas
Writable
Smart Insert
17 : 24 : 209

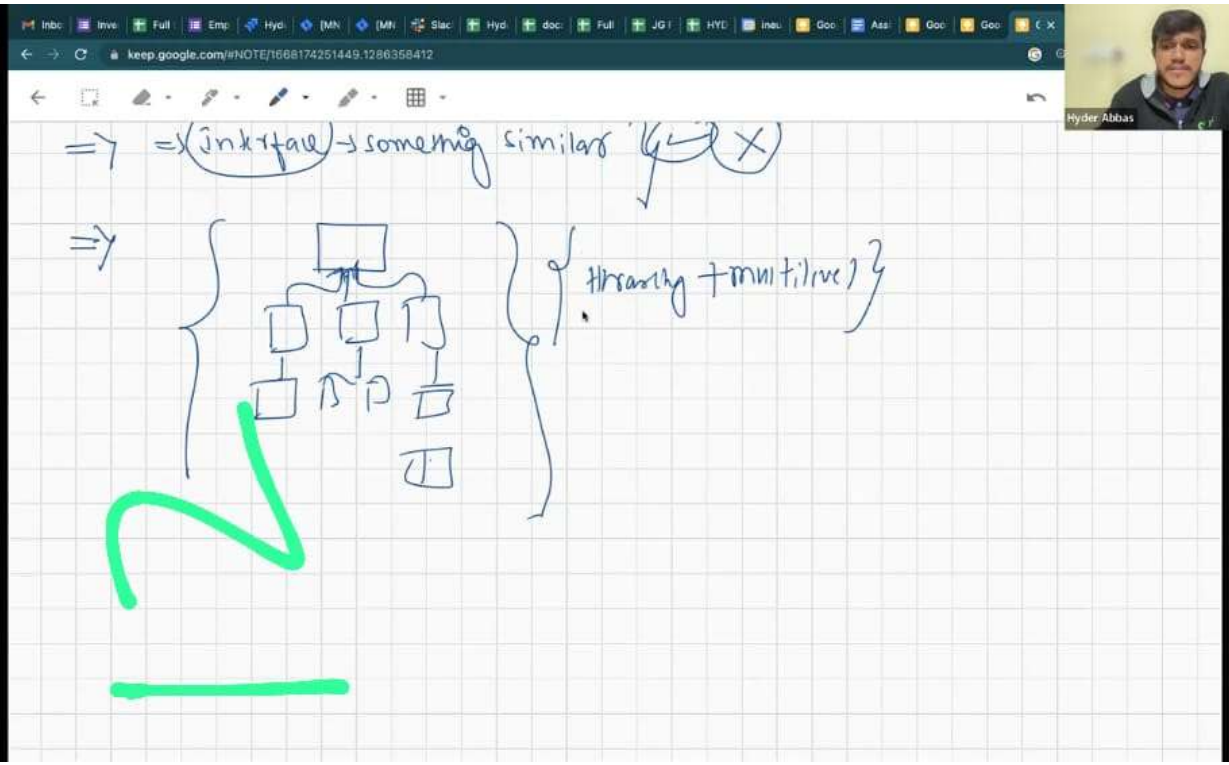

Package Explorer
LaunchDog.java
LaunchStd.java
LaunchLoan.java
LaunchInh1.java
Hyder Abbas
class Demo11
{
String name="Hyder";
int age=28;
void disp()
{
System.out.println("Demo1 "+ age + name);
}
}
class Demo12 extends Demo11
{
}
class Demo13 extends Demo12
{
}
public class LaunchMultilevel {
public static void main(String[] args) {
Demo13 d=new Demo13();
d.disp();
Writable
Smart Insert

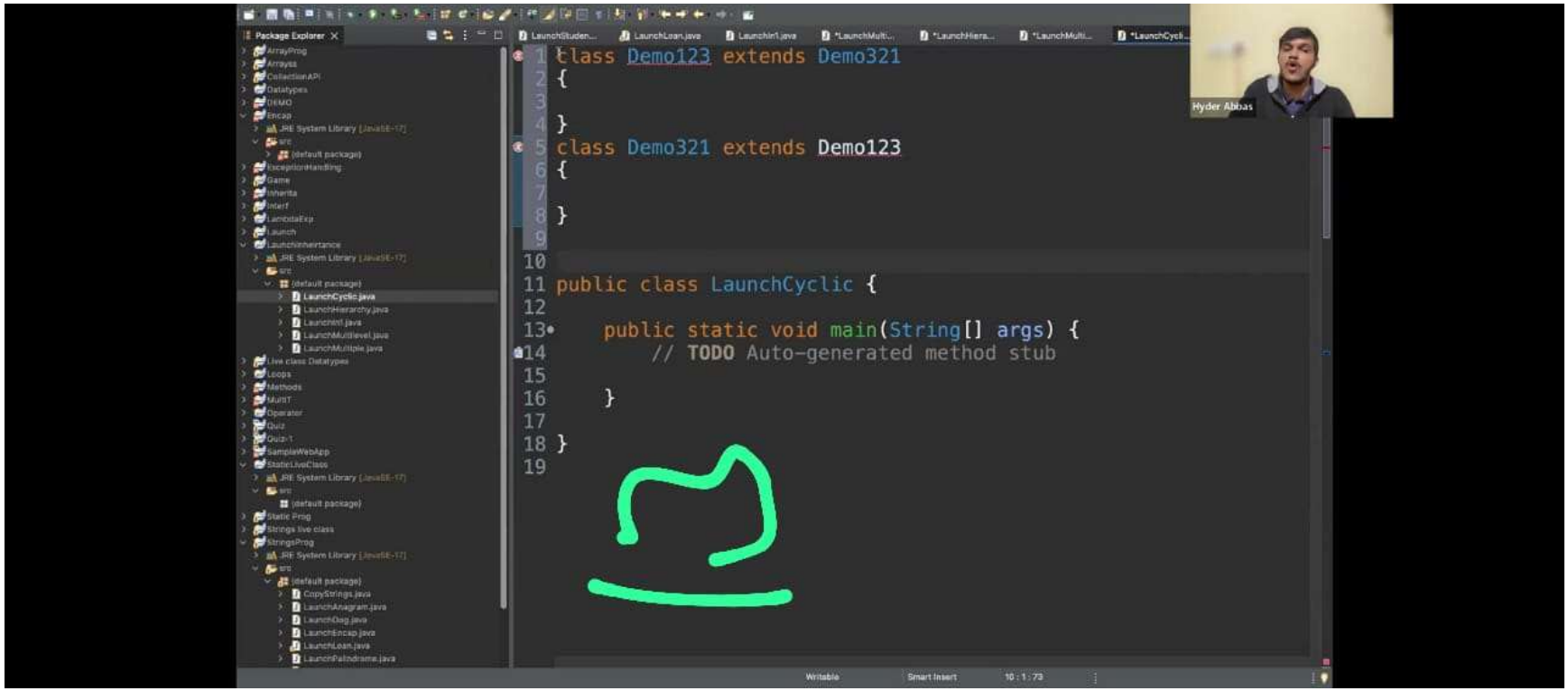

keep.google.com/#NOTE/1668174251449.1286358412
Hyder Abbas
(*) key points => Inheritance (IS-A relationship).
(1) => Single inheritance is allowed (one class can extend another class) ✓
(2) => Object class is parent of all class ; => ✓
(3) => Multilevel Inheritance is allowed ; ✓
Demo1
Demo2
Demo3

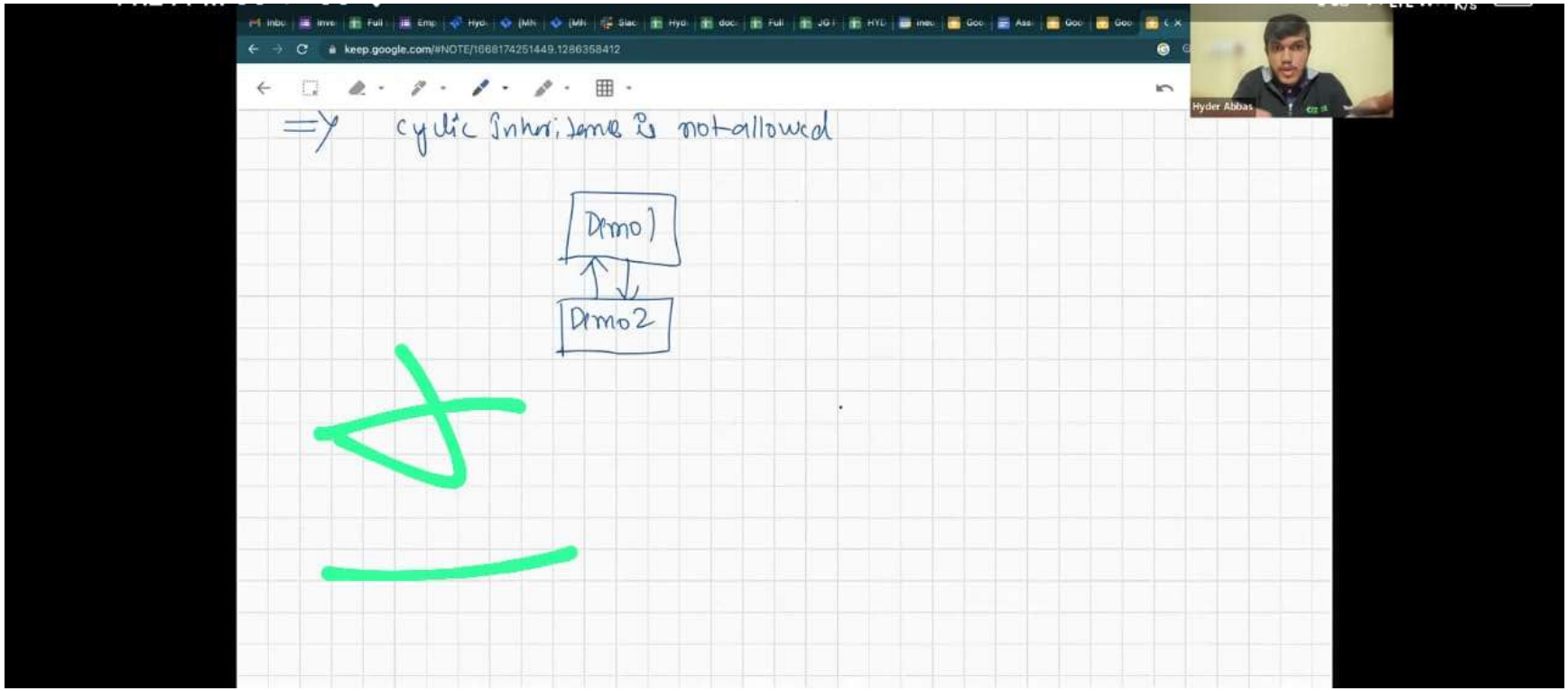

keep.google.com/#NOTE/1668174251449.1286358412
Hybrid
IDOs
=> multiple Inheritance is not allowed
Hyder Abbas

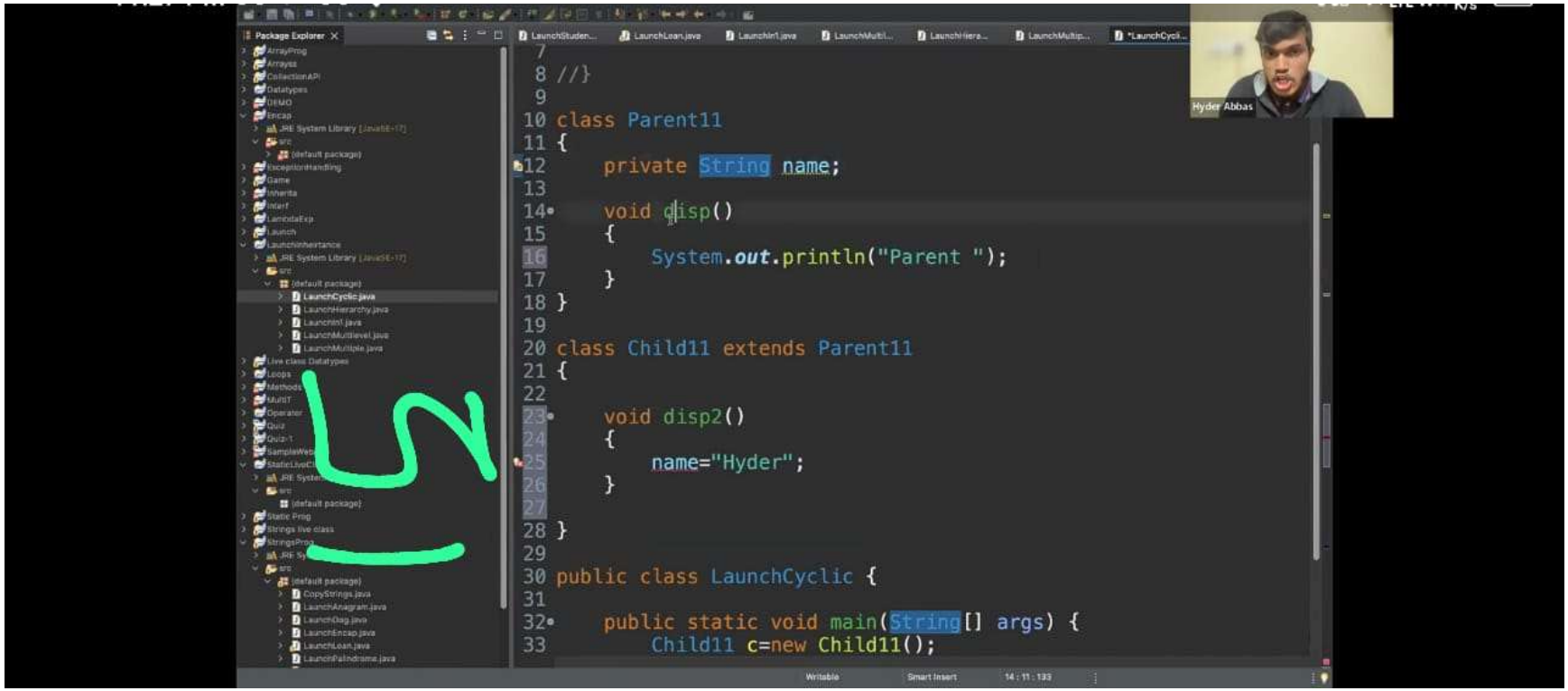

keep.google.com/#NOTE/1668174251449.1286358412
Hyder Abbas
=> multiple Inheritance is not allowed
Object
Demo1
int i=10
Demo2
int i=20
Demo3
ambiguous -> compile

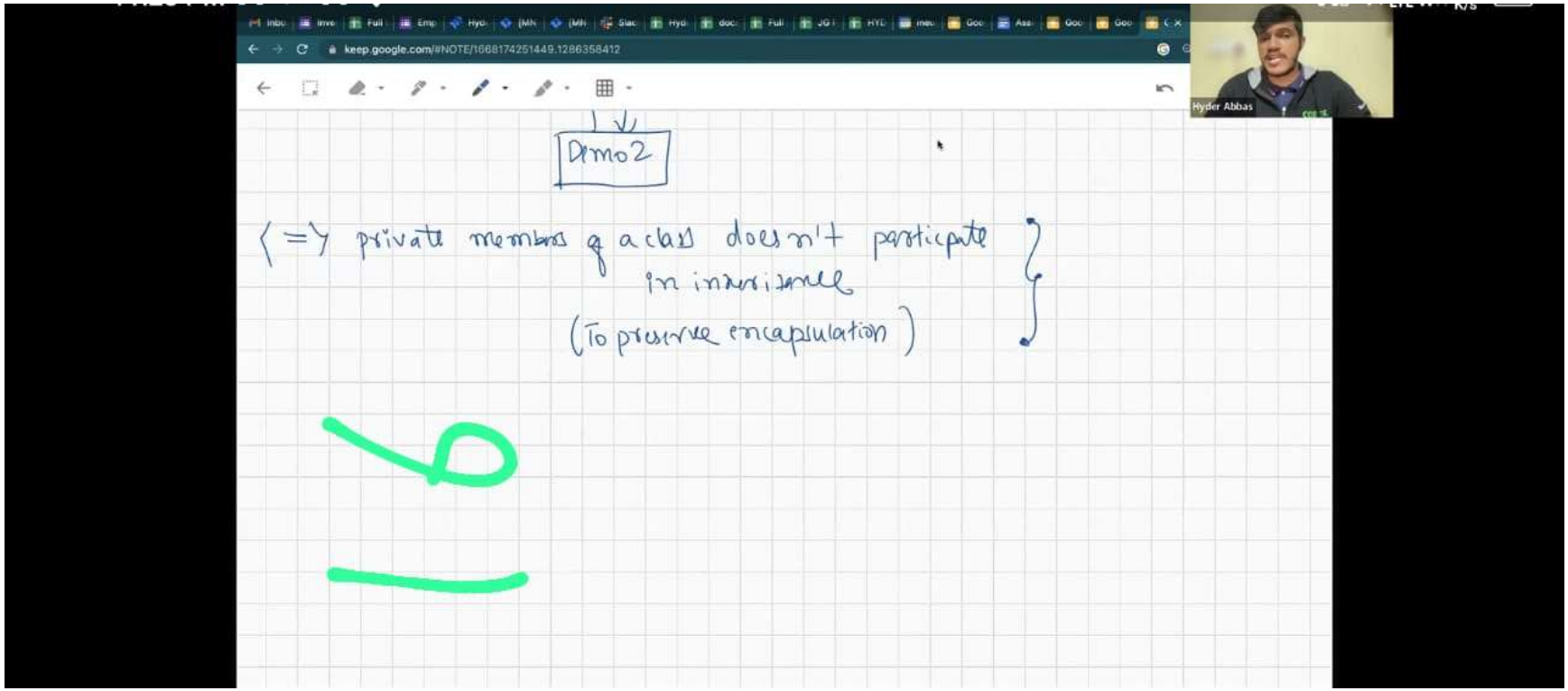

Package Explorer
LaunchStd.java
LaunchStuden...
LaunchLoan.java
LaunchIn1.java
*LaunchMulti...
*LaunchHiera...
*LaunchMulti...
Hyder Abbas
class Parent1
{
}
class Parent2
{
}
//class Child extends Parent1 ,extends Parent2
//{
//}
public class LaunchMultiple {
public static void main(String[] args) {
// TODO Auto-generated method stub
}
}
Writable
Smart Insert
14 : 3 : 97

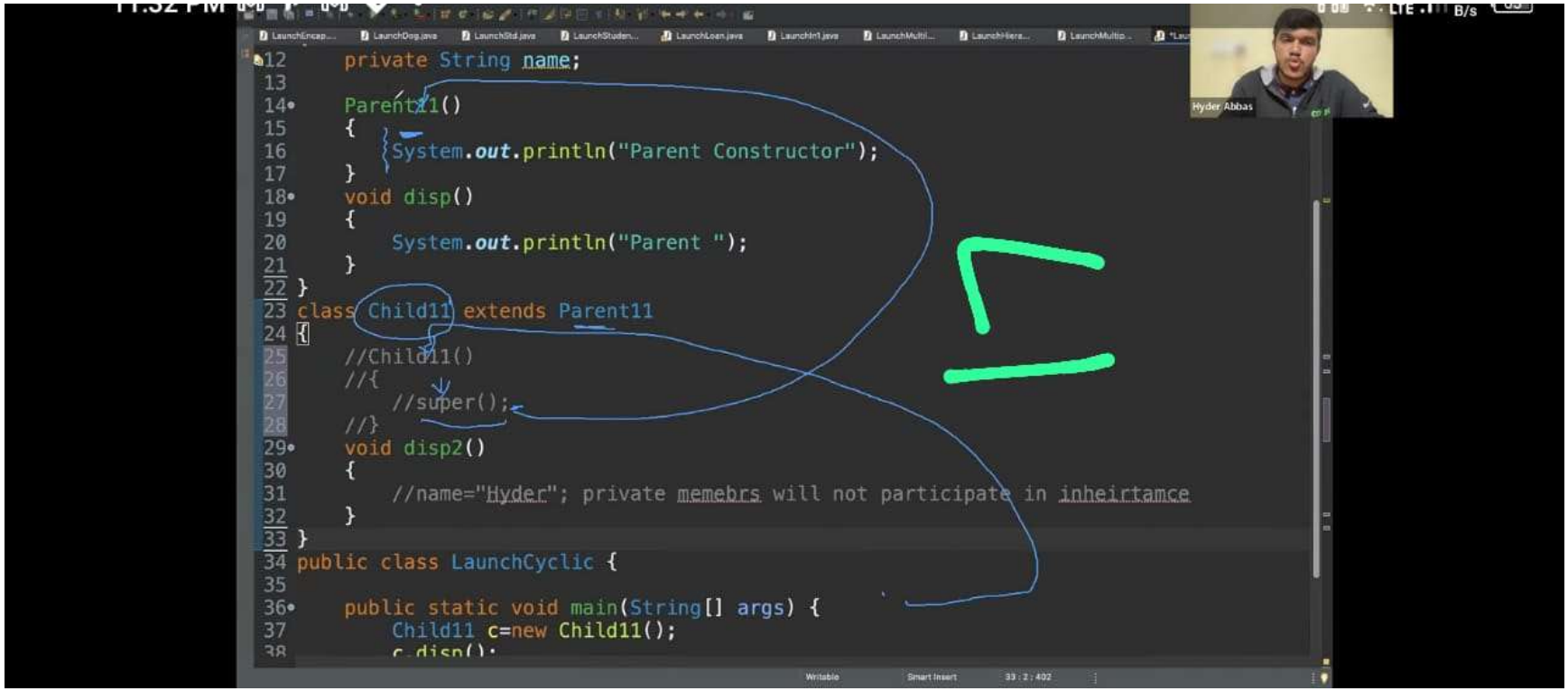

keep.google.com/#NOTE/1668174251449.1286358412
Hyder Abbas
=> (interface) -> something similar
=>
Hierarchy + multilevel

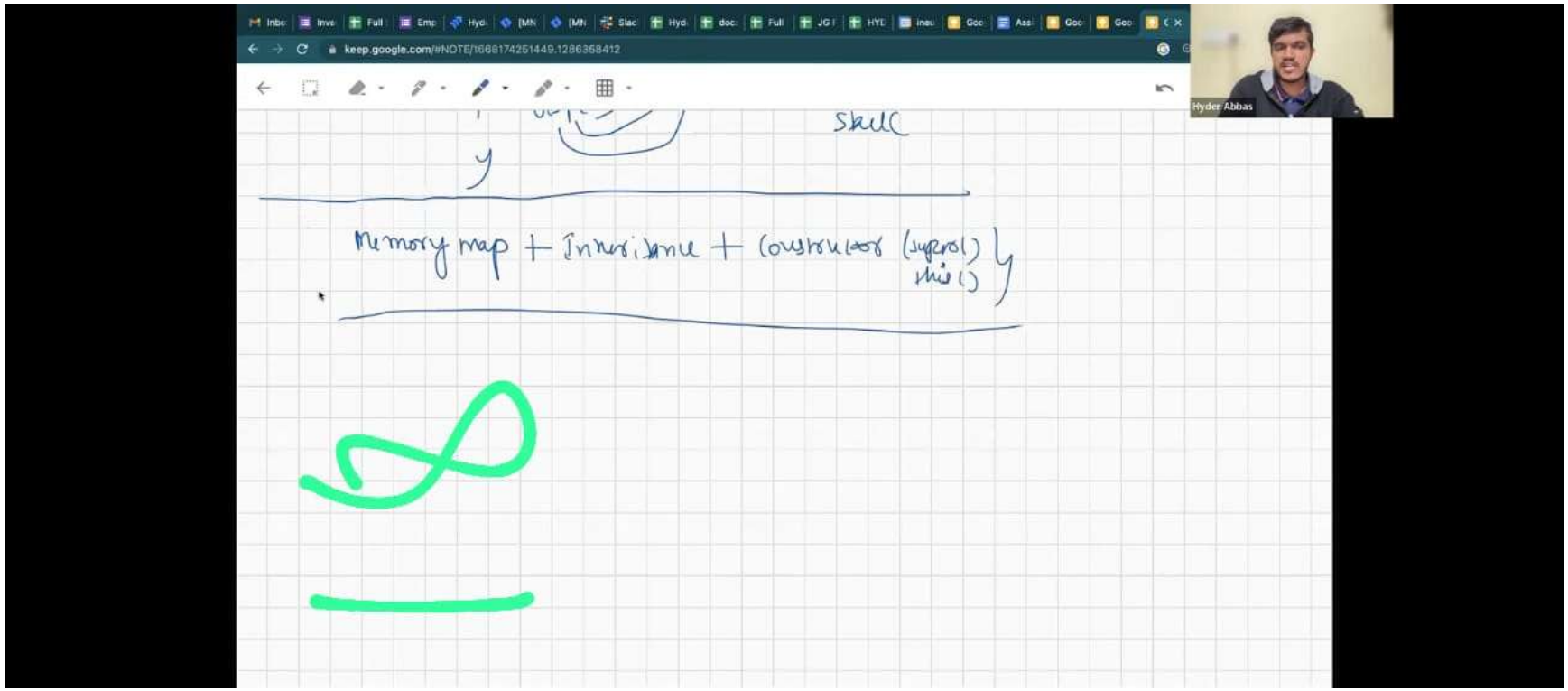

Package Explorer
class Demo123 extends Demo321
{
}
class Demo321 extends Demo123
{
}
public class LaunchCyclic {
public static void main(String[] args) {
// TODO Auto-generated method stub
}
}
Hyder Abbas
Writable
Smart Insert
10 : 1 : 73

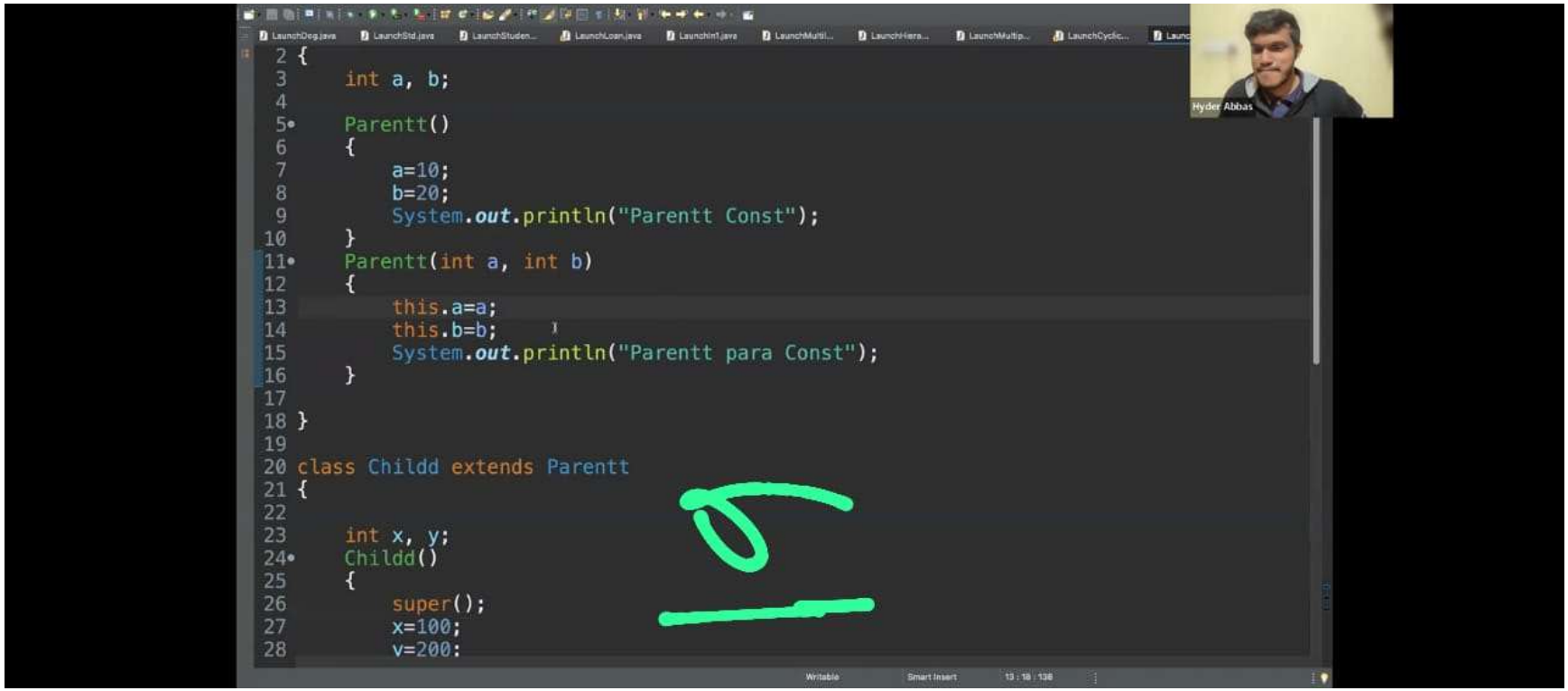

keep.google.com/#NOTE/1668174251449.1286358412
=> cyclic Inheritance is not allowed
Demo1
Demo2
Hyder Abbas

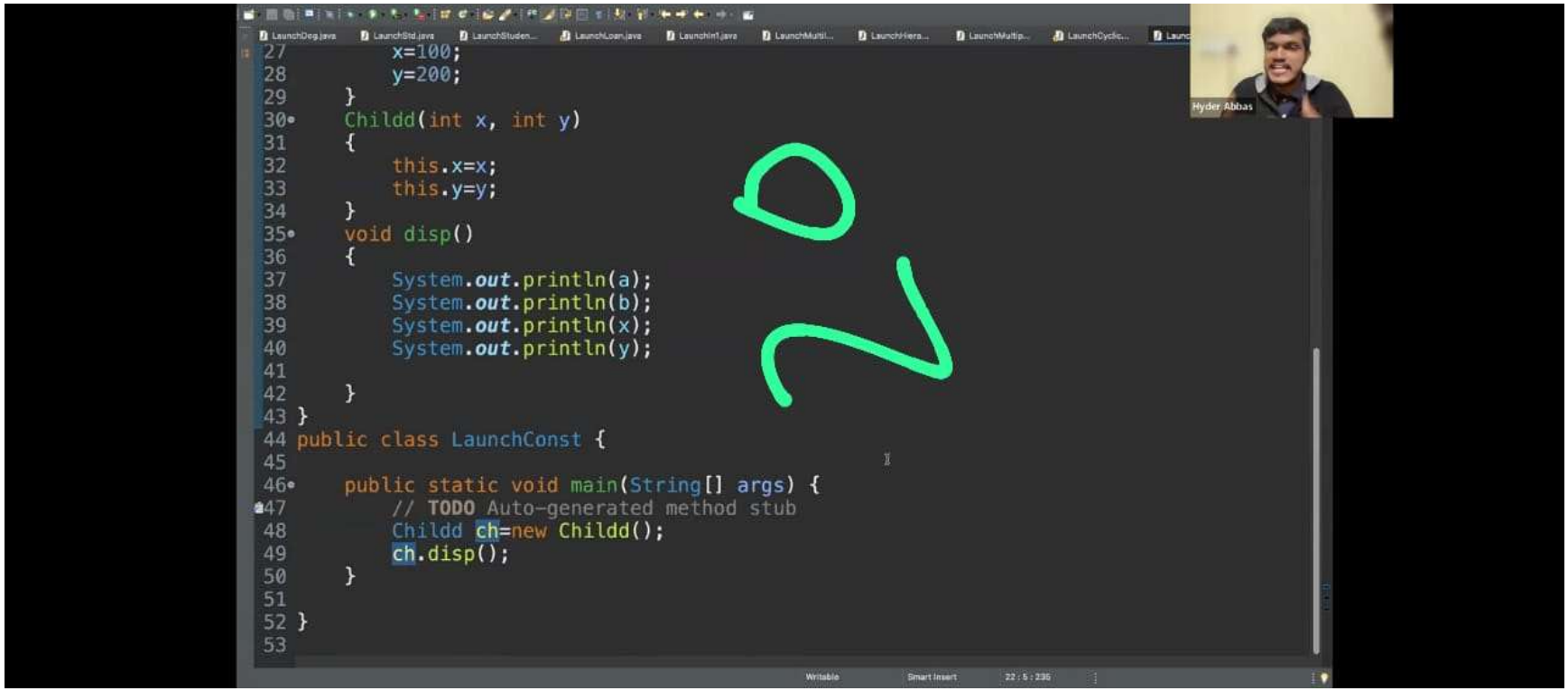

Package Explorer
LaunchCyclic.java
LaunchHierarchy.java
LaunchInh1.java
LaunchMultilevel.java
LaunchMultiple.java
CopyStrings.java
LaunchAnagram.java
LaunchDiag.java
LaunchEncap.java
LaunchLoan.java
LaunchPalindrome.java
//}
class Parent11
{
private String name;
void disp()
{
System.out.println("Parent ");
}
}
class Child11 extends Parent11
{
void disp2()
{
name="Hyder";
}
}
public class LaunchCyclic {
public static void main(String[] args) {
Child11 c=new Child11();
Hyder Abbas
Writable
Smart Insert
14 : 11 : 133

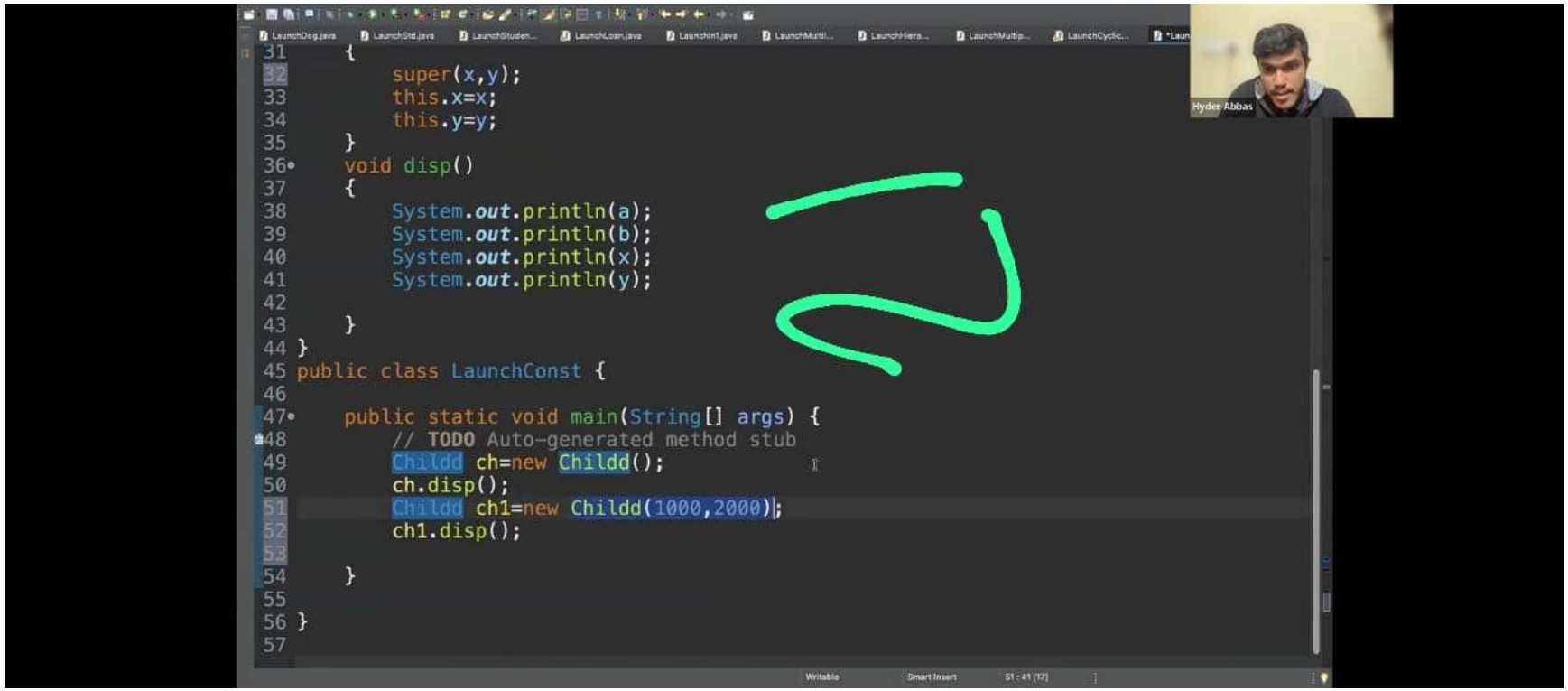

keep.google.com/#NOTE/1668174251449.1286358412
Demo2
⟨=⟩ private members of a class doesn't participate in inheritance
(To preserve encapsulation)
Hyder Abbas

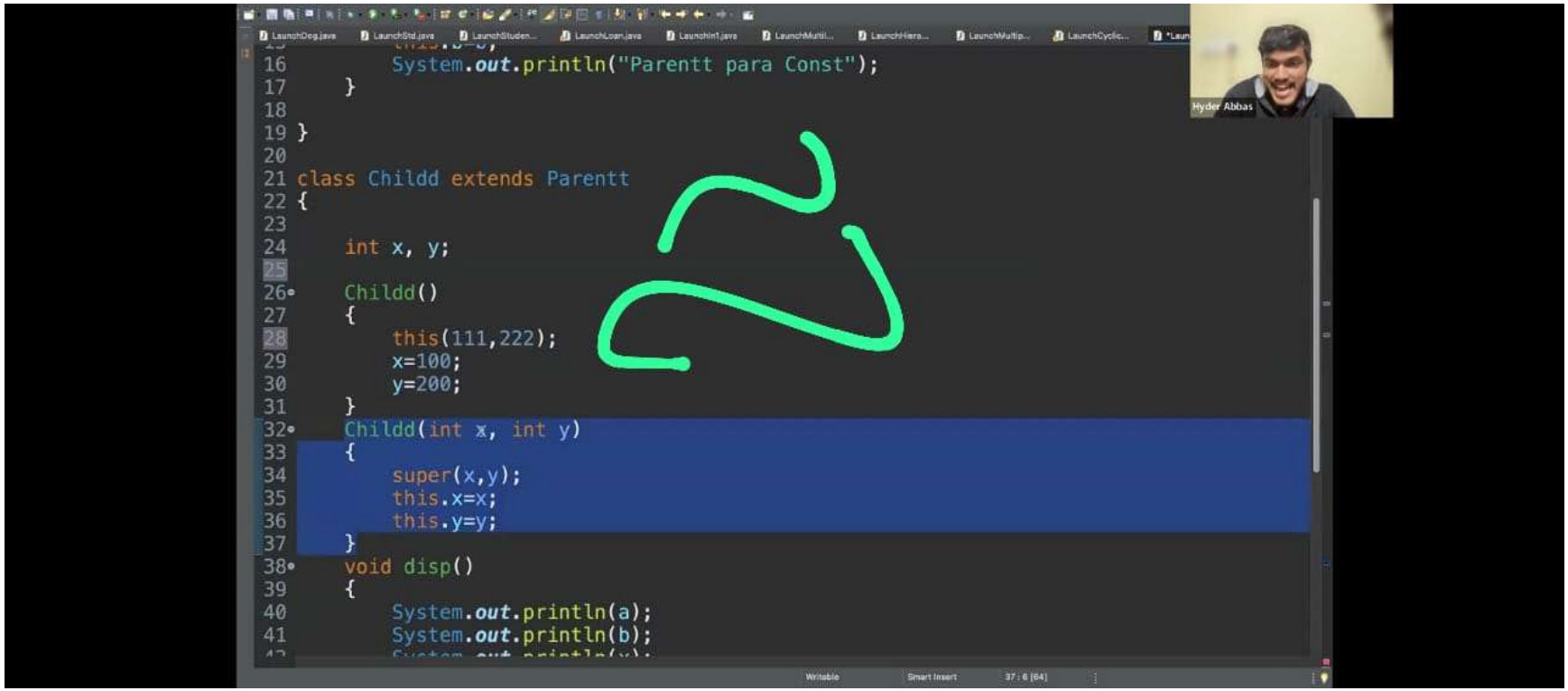

Hyder Abbas
12 private String name;
13
14 Parent11()
15 {
16 System.out.println("Parent Constructor");
17 }
18 void disp()
19 {
20 System.out.println("Parent ");
21 }
22 }
23 class Child11 extends Parent11
24 {
25 //Child11()
26 //{
27 //super();
28 //}
29 void disp2()
30 {
31 //name="Hyder"; private memebrs will not participate in inheirtamce
32 }
33 }
34 public class LaunchCyclic {
35
36 public static void main(String[] args) {
37 Child11 c=new Child11();
38 c.disp();
Writable
Smart Insert
33 : 2 : 402

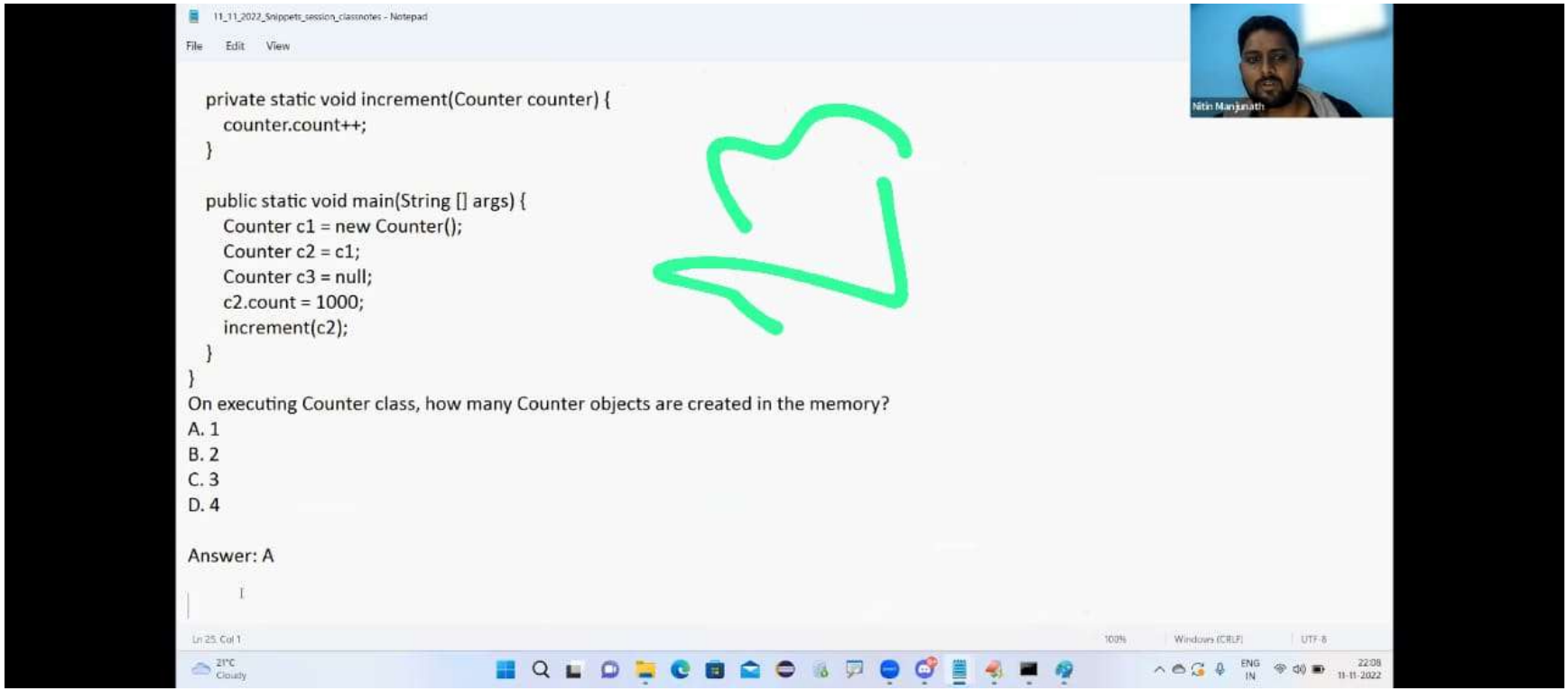

keep.google.com/#NOTE/1668174251449.1286358412
Skull
Memory map + Inheritance + Constructor (super() this())
Hyder Abbas

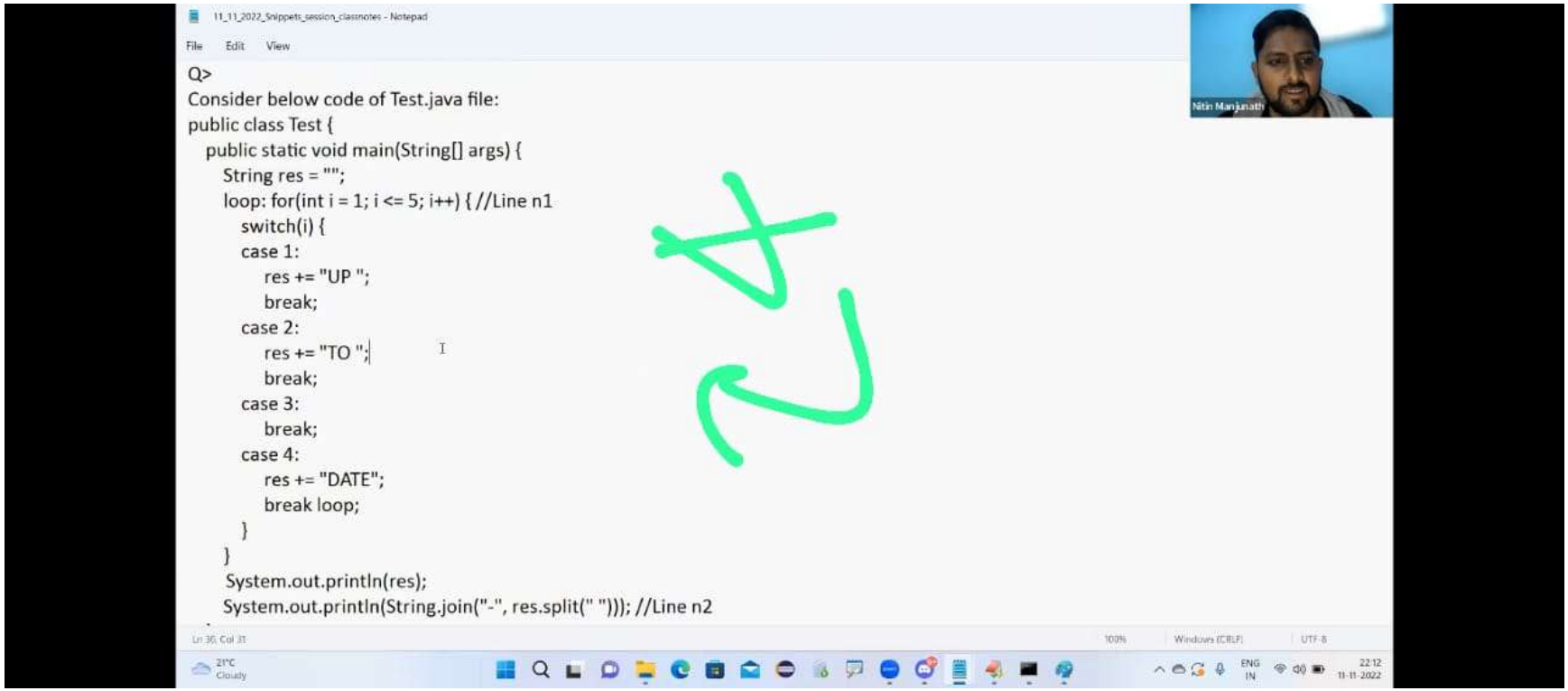

Hyder Abbas
{
int a, b;
Parentt()
{
a=10;
b=20;
System.out.println("Parentt Const");
}
Parentt(int a, int b)
{
this.a=a;
this.b=b;
System.out.println("Parentt para Const");
}
}
class Childd extends Parentt
{
int x, y;
Childd()
{
super();
x=100;
v=200;
Writable
Smart Insert

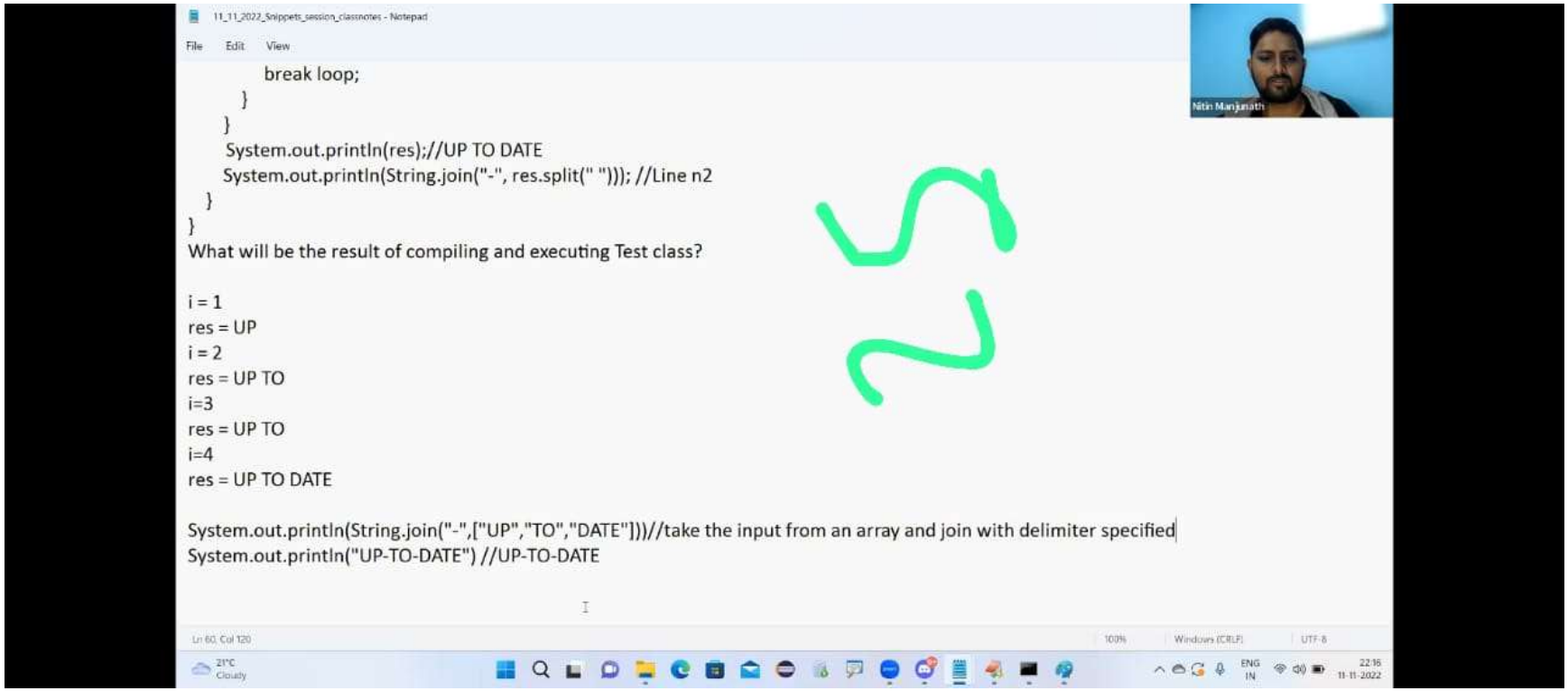

x=100;
y=200;
}
Childd(int x, int y)
{
this.x=x;
this.y=y;
}
void disp()
{
System.out.println(a);
System.out.println(b);
System.out.println(x);
System.out.println(y);
}
}
public class LaunchConst {
public static void main(String[] args) {
// TODO Auto-generated method stub
Childd ch=new Childd();
ch.disp();
}
}
2D
Hyder Abbas

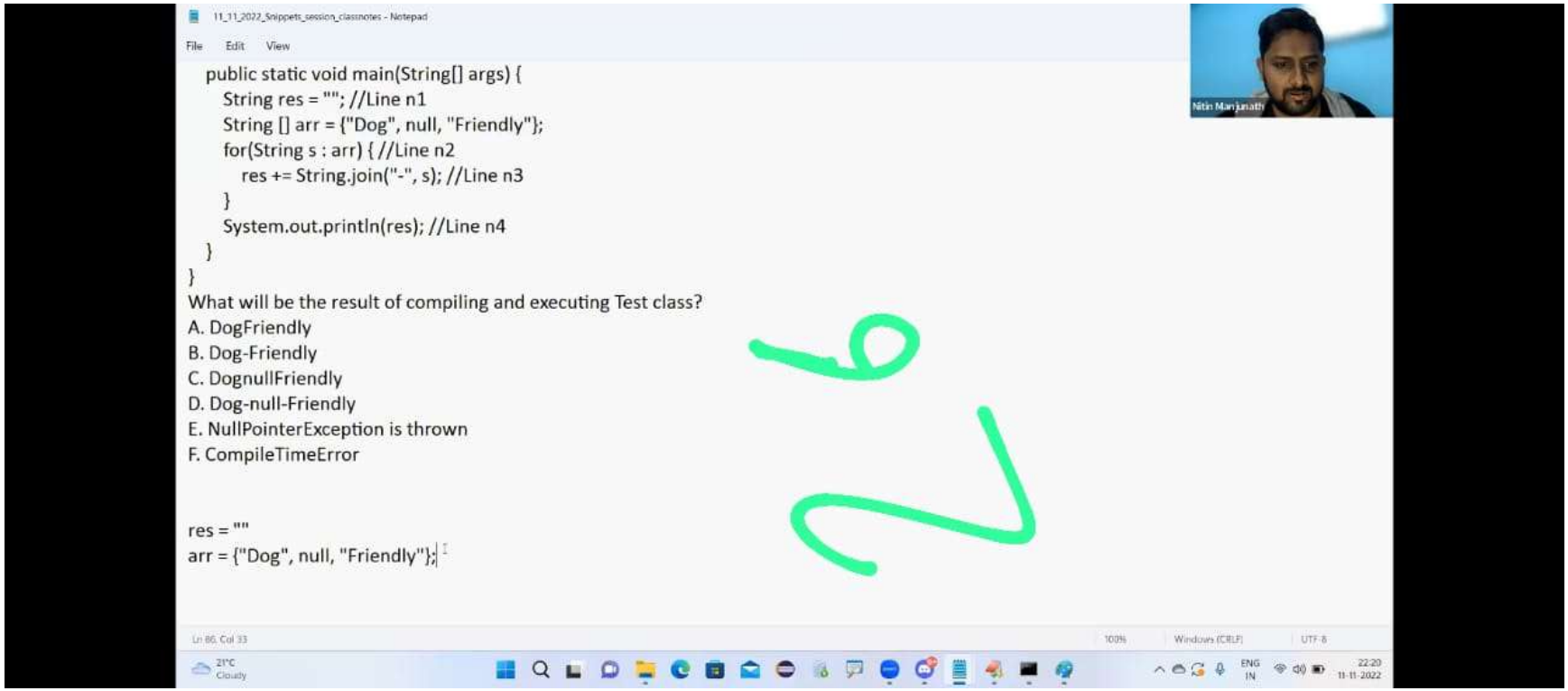

LaunchDog.java
LaunchStd.java
LaunchStuden...
LaunchLoan.java
LaunchInt.java
LaunchMulti...
LaunchHiera...
LaunchMultip...
LaunchCyclic...
*Laun
Hyder Abbas
31 {
32 super(x,y);
33 this.x=x;
34 this.y=y;
35 }
36 void disp()
37 {
38 System.out.println(a);
39 System.out.println(b);
40 System.out.println(x);
41 System.out.println(y);
42
43 }
44 }
45 public class LaunchConst {
46
47 public static void main(String[] args) {
48 // TODO Auto-generated method stub
49 Childd ch=new Childd();
50 ch.disp();
51 Childd ch1=new Childd(1000,2000);
52 ch1.disp();
53
54 }
55
56 }
57
Writable
Smart Insert
51 : 41 [17]

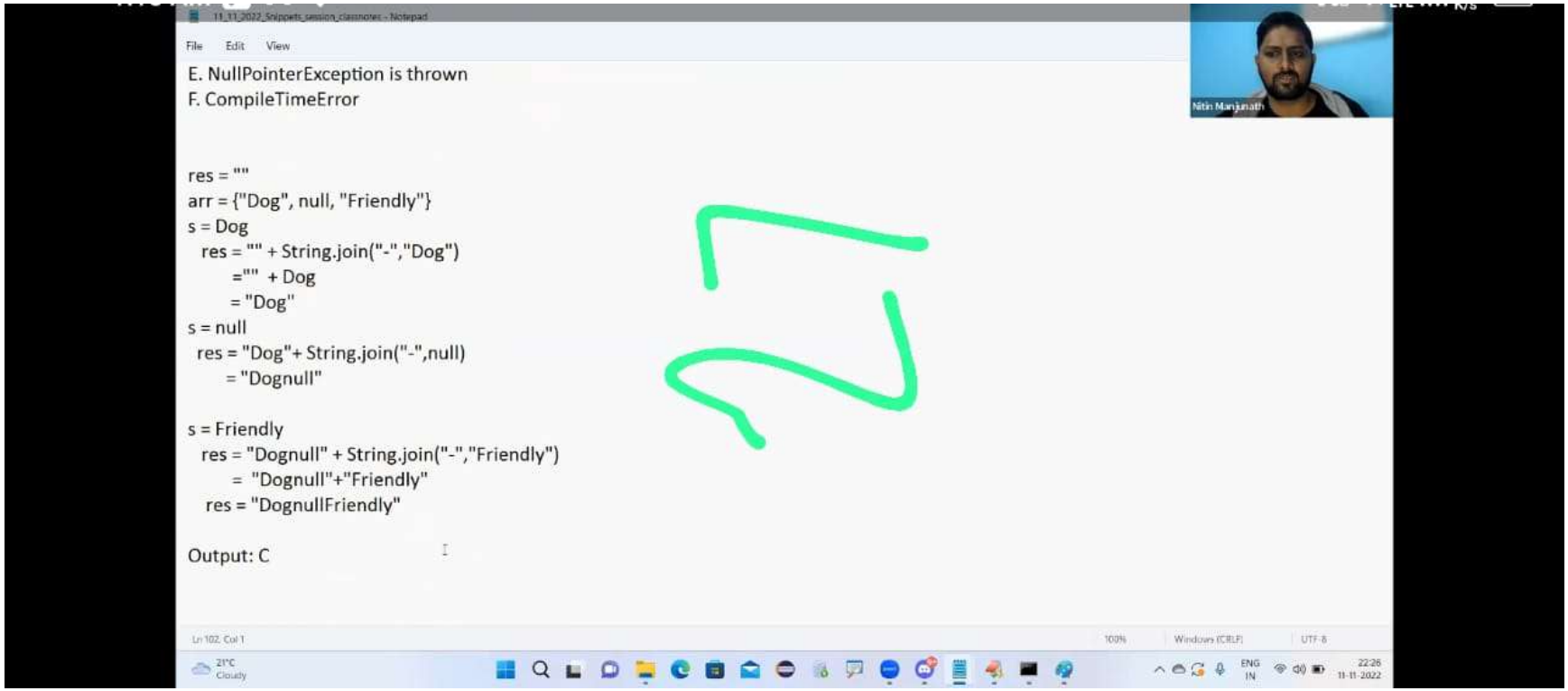

System.out.println("Parentt para Const");
}
}
class Childd extends Parentt
{
int x, y;
Childd()
{
this(111,222);
x=100;
y=200;
}
Childd(int x, int y)
{
super(x,y);
this.x=x;
this.y=y;
}
void disp()
{
System.out.println(a);
System.out.println(b);
Hyder Abbas
Writable
Smart Insert
37 : 6 [64]

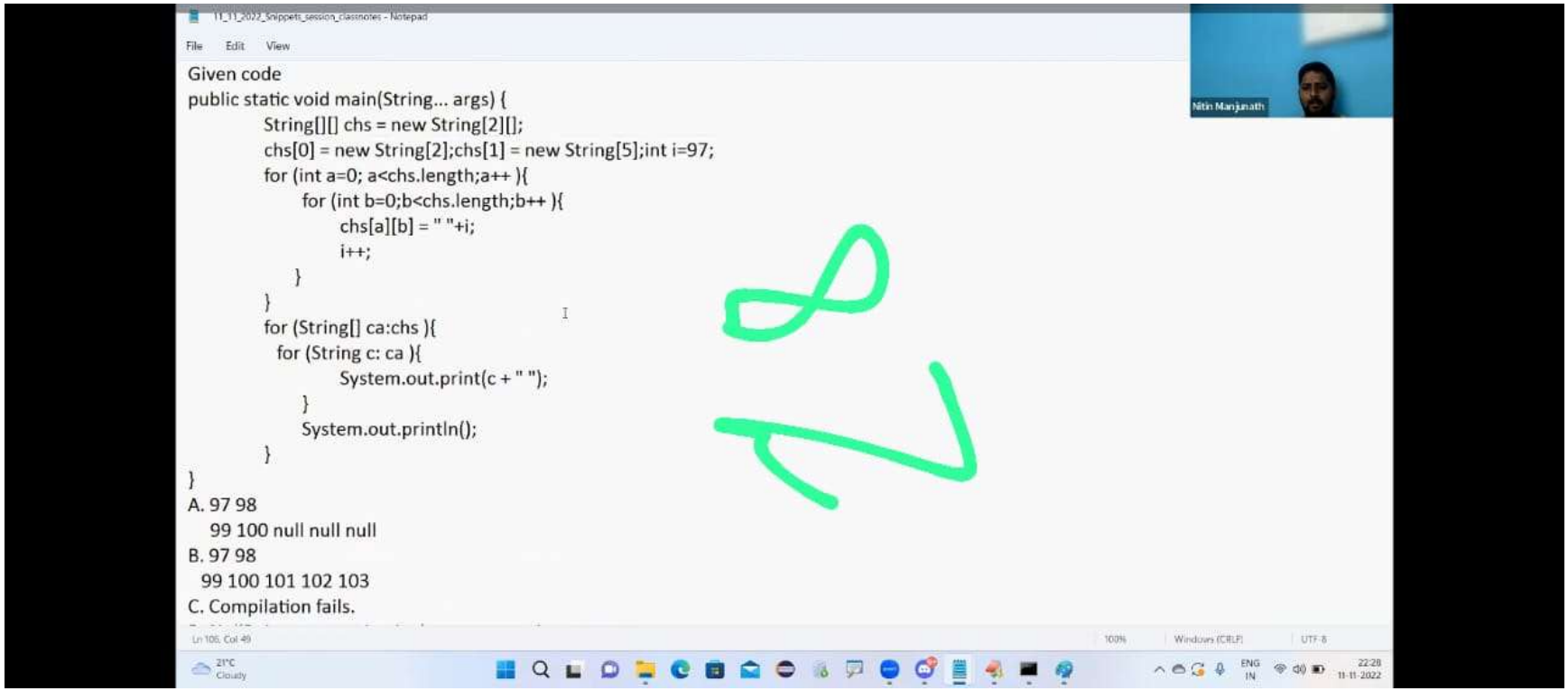

11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
private static void increment(Counter counter) {
counter.count++;
}
public static void main(String [] args) {
Counter c1 = new Counter();
Counter c2 = c1;
Counter c3 = null;
c2.count = 1000;
increment(c2);
}
}
On executing Counter class, how many Counter objects are created in the memory?
A. 1
B. 2
C. 3
D. 4
Answer: A
Nitin Manjunath
Ln 25, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Cloudy
ENG
IN
22:08
11-11-2022

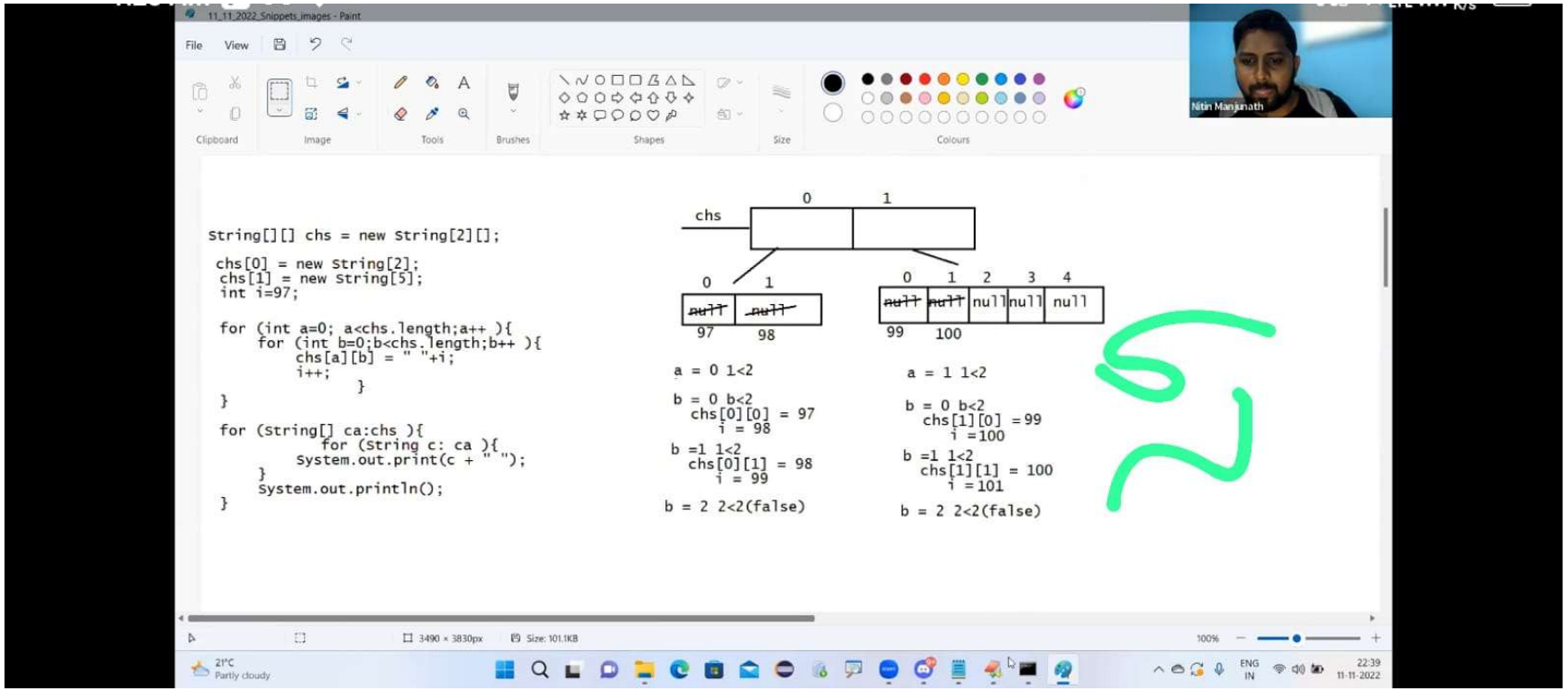

11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
Q>
Consider below code of Test.java file:
public class Test {
public static void main(String[] args) {
String res = "";
loop: for(int i = 1; i <= 5; i++) { //Line n1
switch(i) {
case 1:
res += "UP ";
break;
case 2:
res += "TO ";
break;
case 3:
break;
case 4:
res += "DATE";
break loop;
}
}
System.out.println(res);
System.out.println(String.join("-", res.split(" "))); //Line n2
Nitin Manjunath

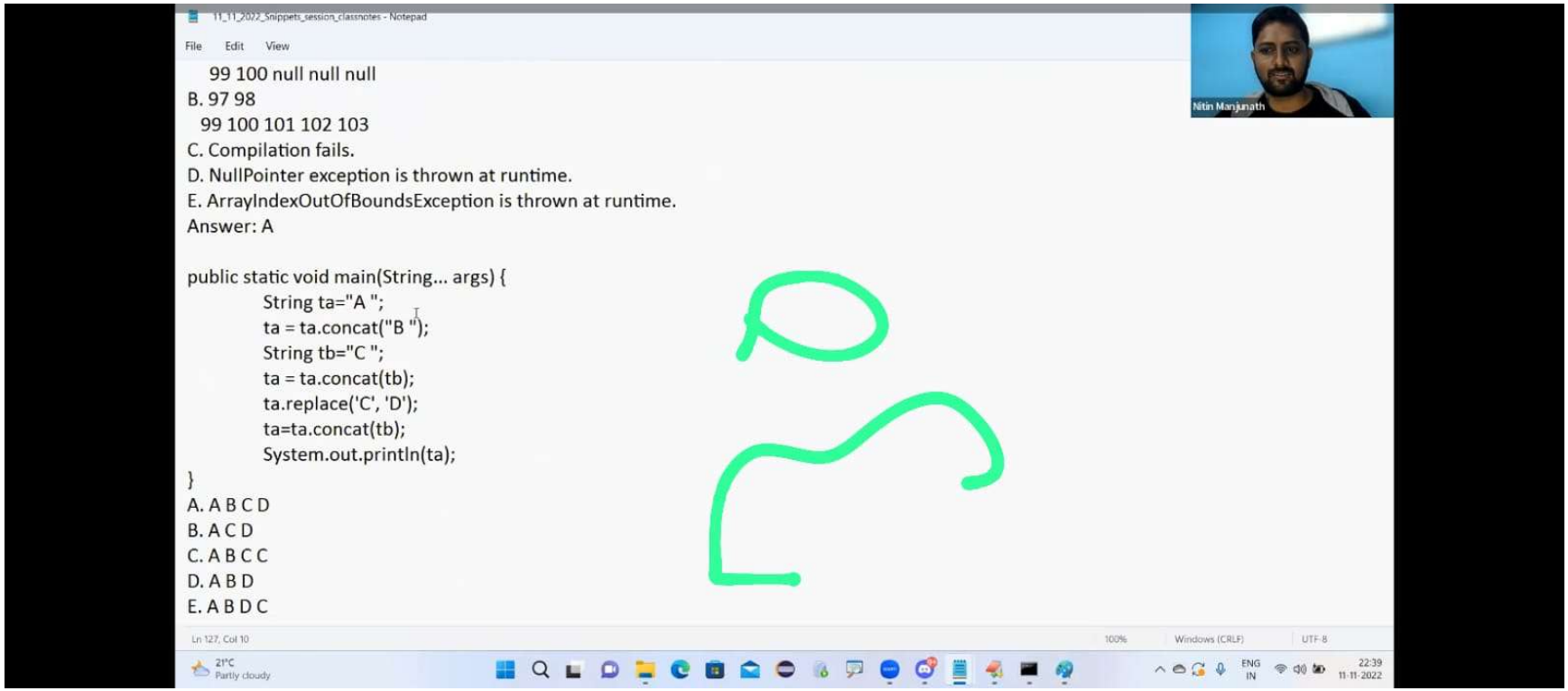

11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
break loop;
}
}
System.out.println(res);//UP TO DATE
System.out.println(String.join("-", res.split(" "))); //Line n2
}
}
What will be the result of compiling and executing Test class?
i = 1
res = UP
i = 2
res = UP TO
i=3
res = UP TO
i=4
res = UP TO DATE
System.out.println(String.join("-",["UP","TO","DATE"]))//take the input from an array and join with delimiter specified
System.out.println("UP-TO-DATE") //UP-TO-DATE
Nitin Manjunath

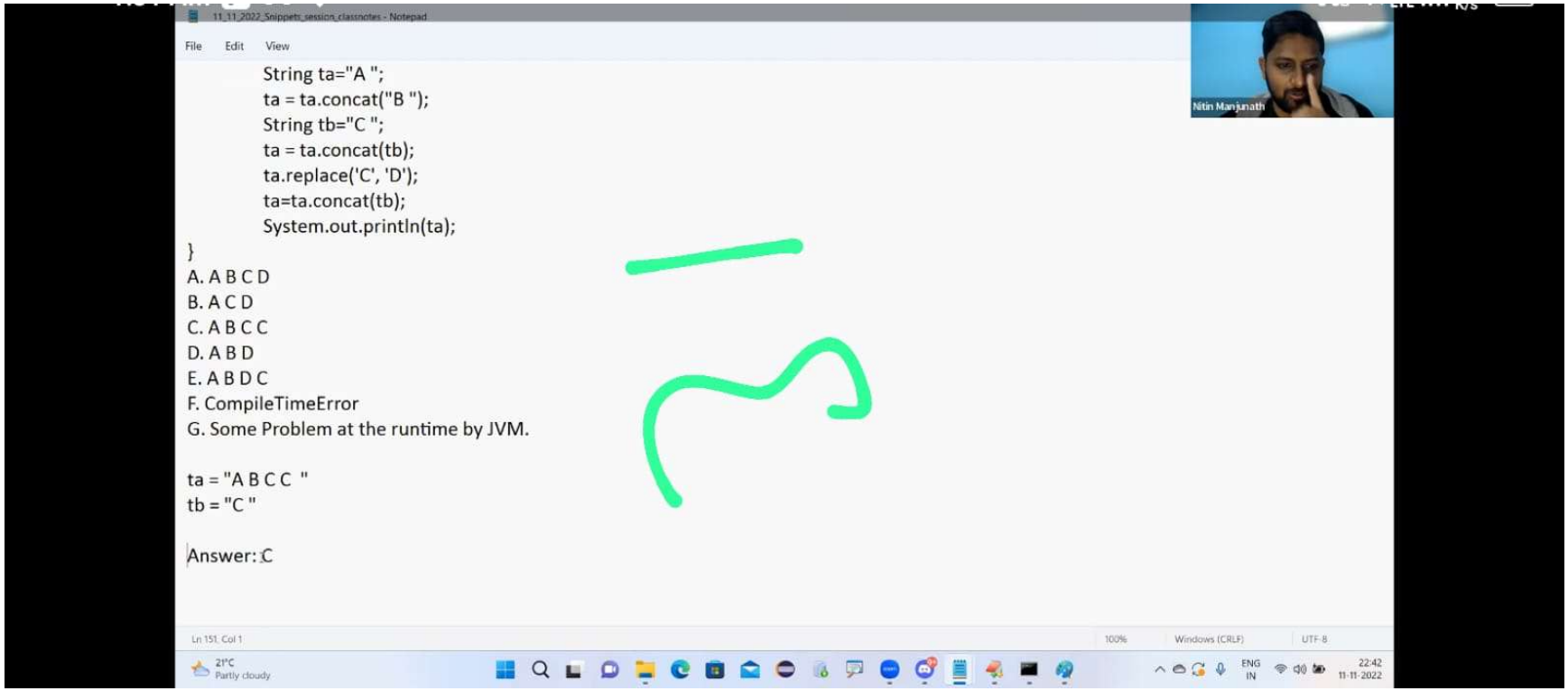

11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
public static void main(String[] args) {
String res = ""; //Line n1
String [] arr = {"Dog", null, "Friendly"};
for(String s : arr) { //Line n2
res += String.join("-", s); //Line n3
}
System.out.println(res); //Line n4
}
}
What will be the result of compiling and executing Test class?
A. DogFriendly
B. Dog-Friendly
C. DognullFriendly
D. Dog-null-Friendly
E. NullPointerException is thrown
F. CompileTimeError
res = ""
arr = {"Dog", null, "Friendly"};
Nitin Manjunath
Ln 86, Col 33
100%
Windows (CRLF)
UTF-8
21°C
Cloudy
ENG
IN
22:20
11-11-2022

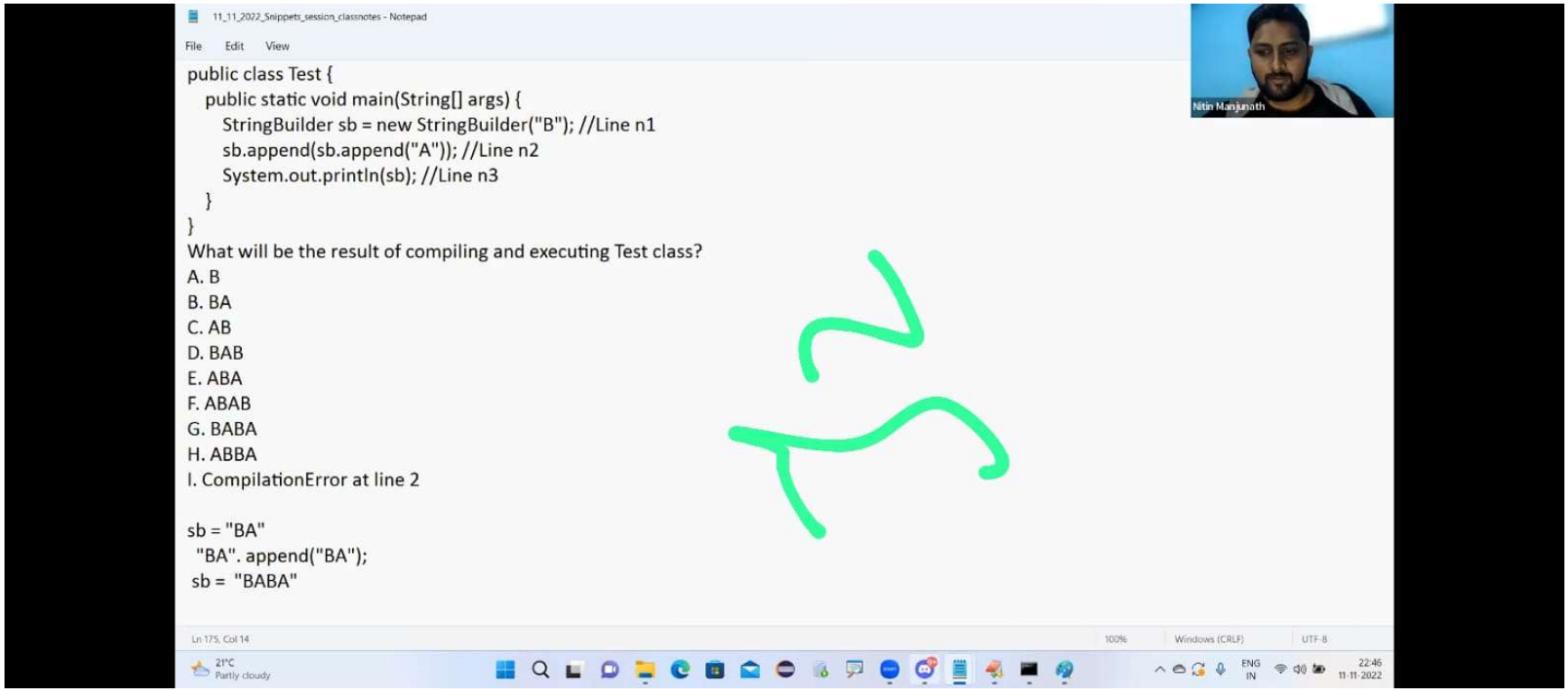

11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
E. NullPointerException is thrown
F. CompileTimeError
res = ""
arr = {"Dog", null, "Friendly"}
s = Dog
res = "" + String.join("-","Dog")
="" + Dog
= "Dog"
s = null
res = "Dog"+ String.join("-",null)
= "Dognull"
s = Friendly
res = "Dognull" + String.join("-","Friendly")
= "Dognull"+"Friendly"
res = "DognullFriendly"
Output: C
Nitin Manjunath
Ln 102, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Cloudy
ENG
IN
22:26
11-11-2022


11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
Given code
public static void main(String... args) {
String[][] chs = new String[2][];
chs[0] = new String[2];chs[1] = new String[5];int i=97;
for (int a=0; a<chs.length;a++){
for (int b=0;b<chs.length;b++){
chs[a][b] = " "+i;
i++;
}
}
for (String[] ca:chs){
for (String c: ca){
System.out.print(c + " ");
}
System.out.println();
}
}
A. 97 98
99 100 null null null
B. 97 98
99 100 101 102 103
C. Compilation fails.
Nitin Manjunath
Ln 106, Col 49
100%
Windows (CRLF)
UTF-8
21°C
Cloudy
ENG
IN
22:28
11-11-2022


11_11_2022_Snippets_images - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
String[][] chs = new String[2][];
chs[0] = new String[2];
chs[1] = new String[5];
int i=97;
for (int a=0; a<chs.length;a++){
for (int b=0;b<chs.length;b++){
chs[a][b] = " "+i;
i++;
}
}
for (String[] ca:chs){
for (String c: ca){
System.out.print(c + " ");
}
System.out.println();
}
chs
0
1
null
null
97
98
null
null
null
null
null
99
100
a = 0 1<2
b = 0 b<2
chs[0][0] = 97
i = 98
b =1 1<2
chs[0][1] = 98
i = 99
b = 2 2<2(false)
a = 1 1<2
b = 0 b<2
chs[1][0] =99
i =100
b =1 1<2
chs[1][1] = 100
i =101
b = 2 2<2(false)
3490 × 3830px
Size: 101.1KB
100%
21°C
Partly cloudy
ENG
IN
22:39
11-11-2022


11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
99 100 null null null
B. 97 98
99 100 101 102 103
C. Compilation fails.
D. NullPointer exception is thrown at runtime.
E. ArrayIndexOutOfBoundsException is thrown at runtime.
Answer: A
public static void main(String... args) {
String ta="A ";
ta = ta.concat("B ");
String tb="C ";
ta = ta.concat(tb);
ta.replace('C', 'D');
ta=ta.concat(tb);
System.out.println(ta);
}
A. A B C D
B. A C D
C. A B C C
D. A B D
E. A B D C
Nitin Manjunath
Ln 127, Col 10
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
ENG
IN
22:39
11-11-2022


11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
String ta="A ";
ta = ta.concat("B ");
String tb="C ";
ta = ta.concat(tb);
ta.replace('C', 'D');
ta=ta.concat(tb);
System.out.println(ta);
}
A. A B C D
B. A C D
C. A B C C
D. A B D
E. A B D C
F. CompileTimeError
G. Some Problem at the runtime by JVM.
ta = "A B C C "
tb = "C "
Answer: C
Nitin Manjunath
Ln 151, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
ENG
IN
22:42
11-11-2022


11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
public class Test {
public static void main(String[] args) {
StringBuilder sb = new StringBuilder("B"); //Line n1
sb.append(sb.append("A")); //Line n2
System.out.println(sb); //Line n3
}
}
What will be the result of compiling and executing Test class?
A. B
B. BA
C. AB
D. BAB
E. ABA
F. ABAB
G. BABA
H. ABBA
I. CompilationError at line 2
sb = "BA"
"BA". append("BA");
sb = "BABA"
Nitin Manjunath
Ln 175, Col 14
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
ENG
IN
22:46
11-11-2022

11_11_2022_Snippets_session_classnotes - Notepad

File Edit View

```
        System.out.println(arr[place.indexOf("a", 3)]); //Line n1
    }
}
```

What will be the result of compiling and executing Test class?

A. 1st
B. 3rd
C. 4th
D. 5th
E. 2nd
F. RunTimeException
G. Compiletime Error at line n-1

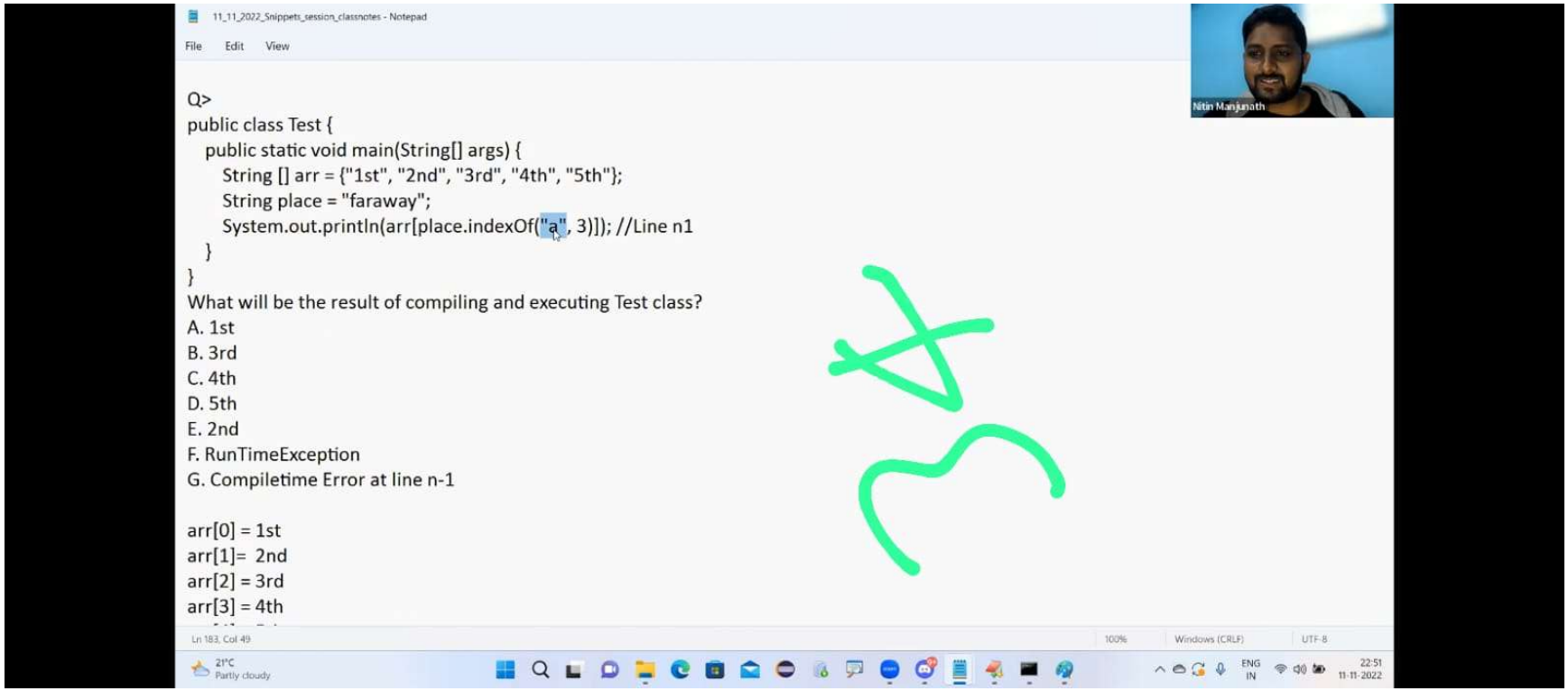
```
arr[0] = 1st
arr[1]=  2nd
arr[2] = 3rd
arr[3] = 4th
arr[4] = 5th
place = "faraway"
 System.out.println( arr[3]);
```

Answer: C

Ln 203, Col 11 100% Windows (CRLF) UTF-8

21°C Partly cloudy

ENG IN 22:50 11-11-2022


11_11_2022_Snippets_session_classnotes - Notepad
File Edit View
Nitin Manjunath
Q>
public class Test {
public static void main(String[] args) {
String [] arr = {"1st", "2nd", "3rd", "4th", "5th"};
String place = "faraway";
System.out.println(arr[place.indexOf("a", 3)]); //Line n1
}
}
What will be the result of compiling and executing Test class?
A. 1st
B. 3rd
C. 4th
D. 5th
E. 2nd
F. RunTimeException
G. Compiletime Error at line n-1
arr[0] = 1st
arr[1]= 2nd
arr[2] = 3rd
arr[3] = 4th
Ln 183, Col 49
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
ENG
IN
22:51
11-11-2022