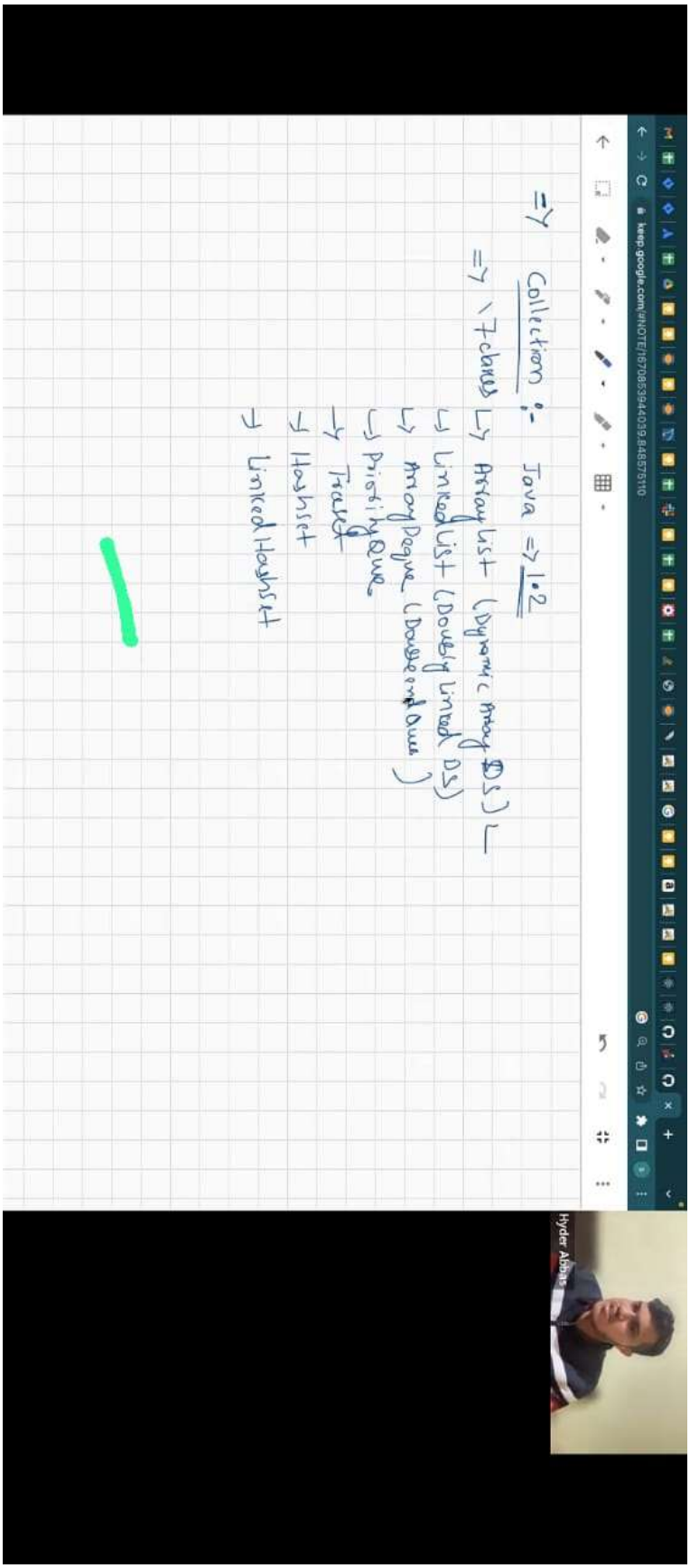
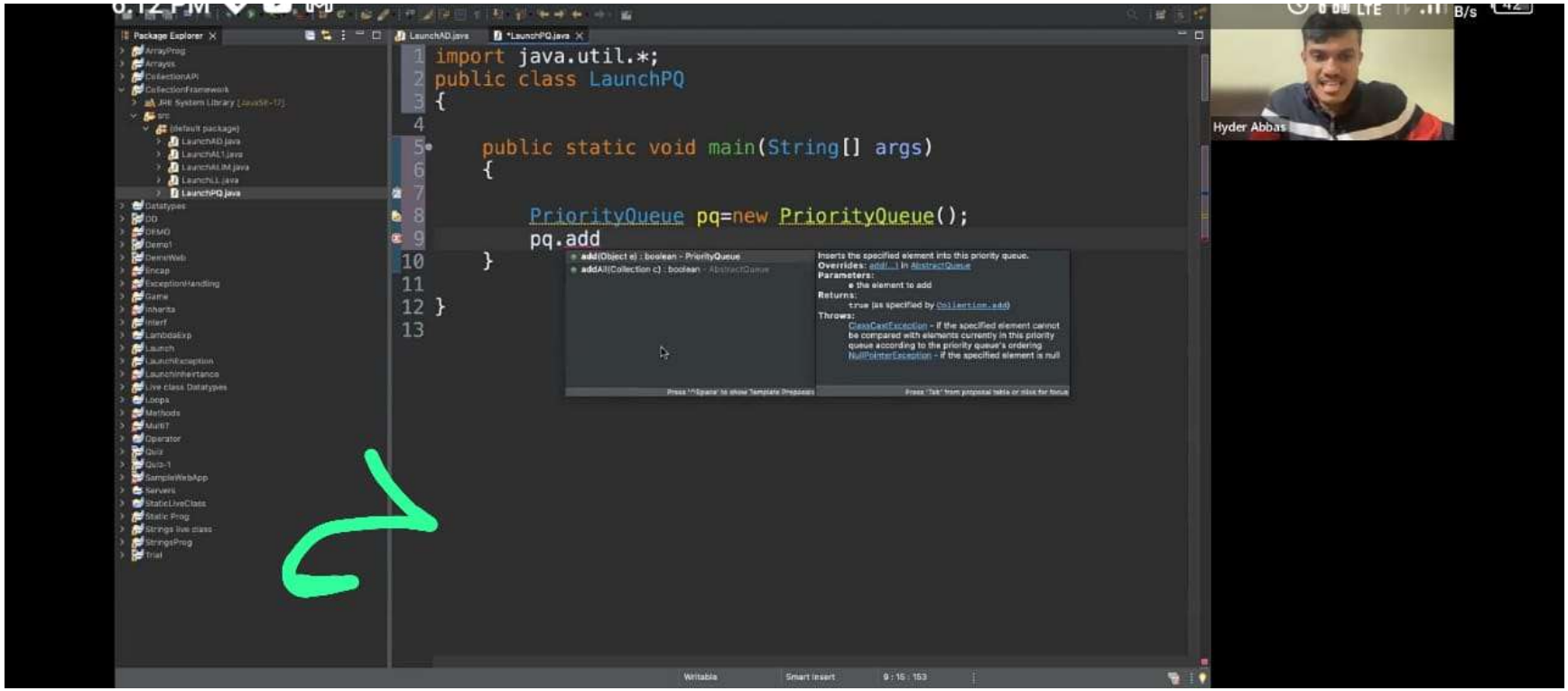
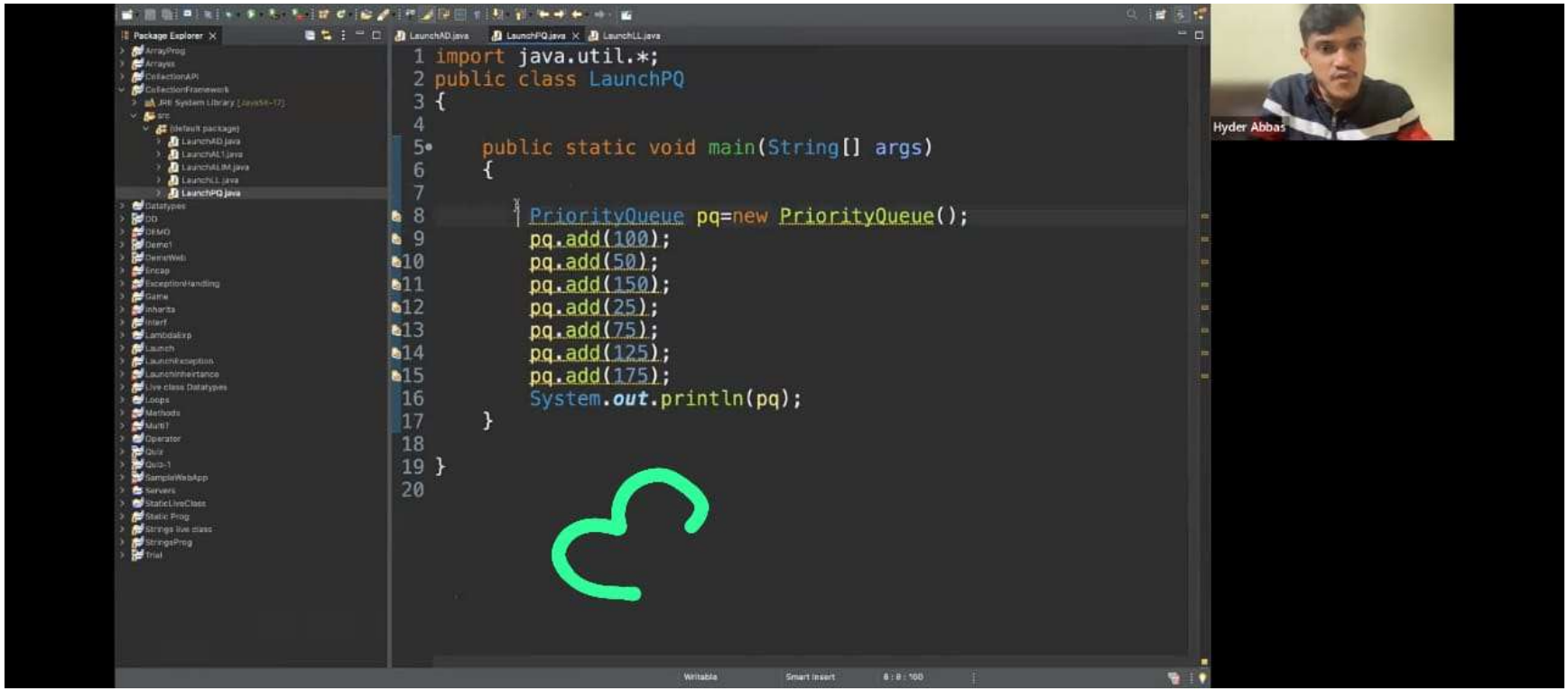
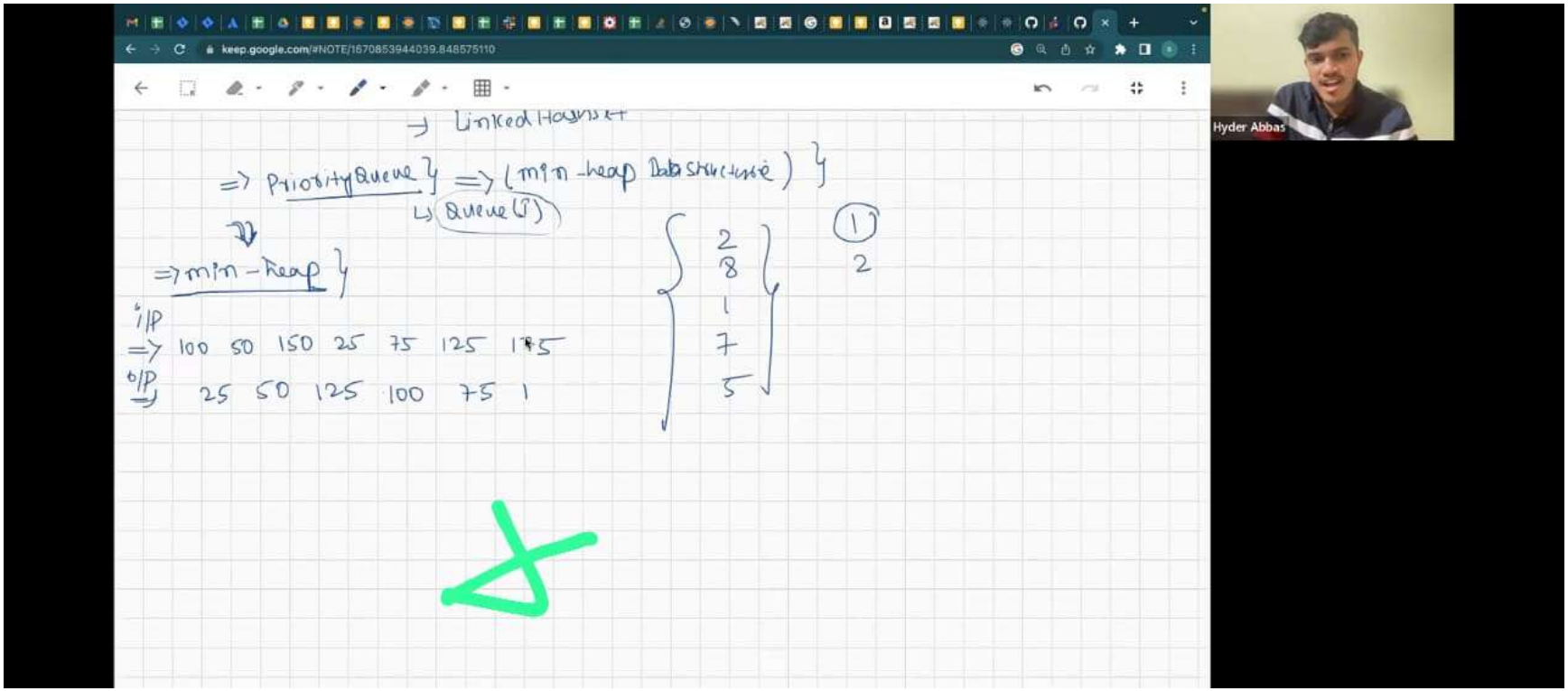
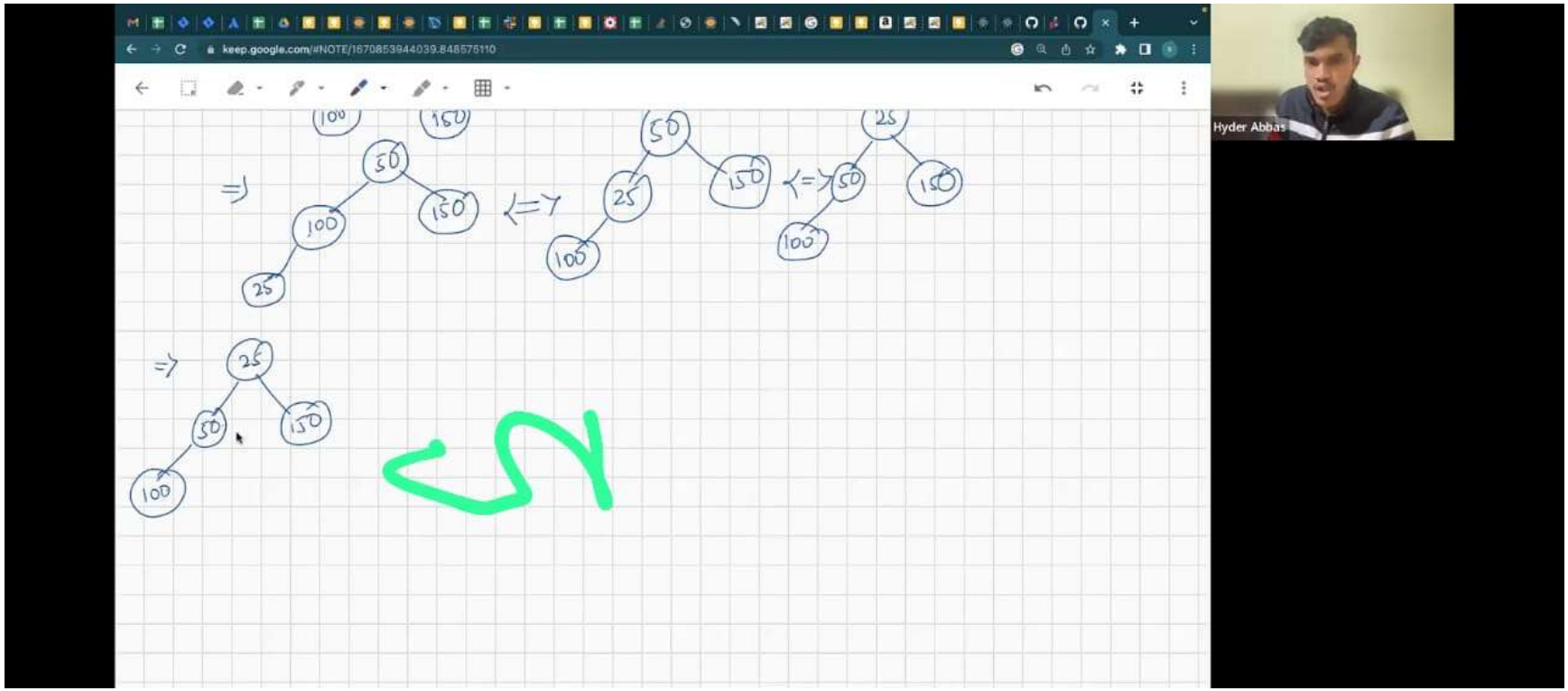
Java Collection Part-2

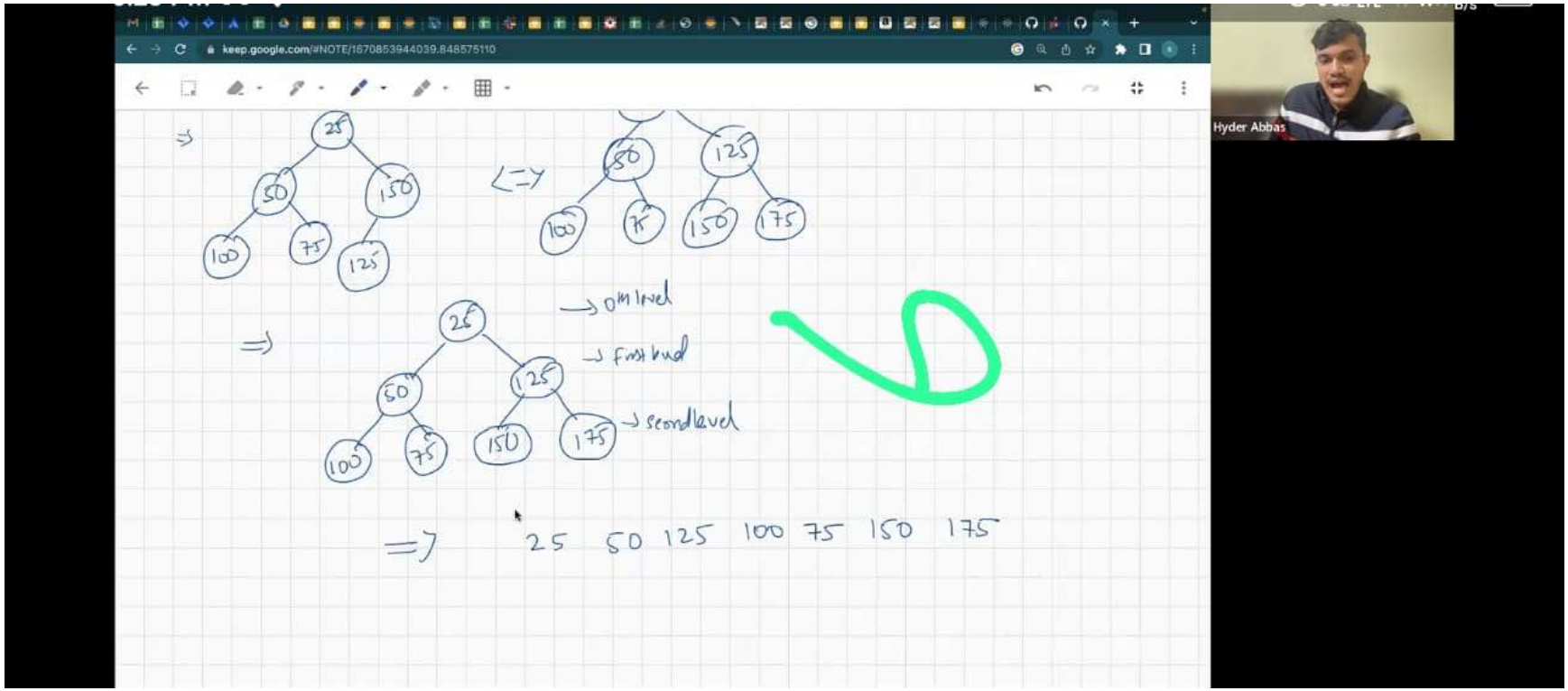


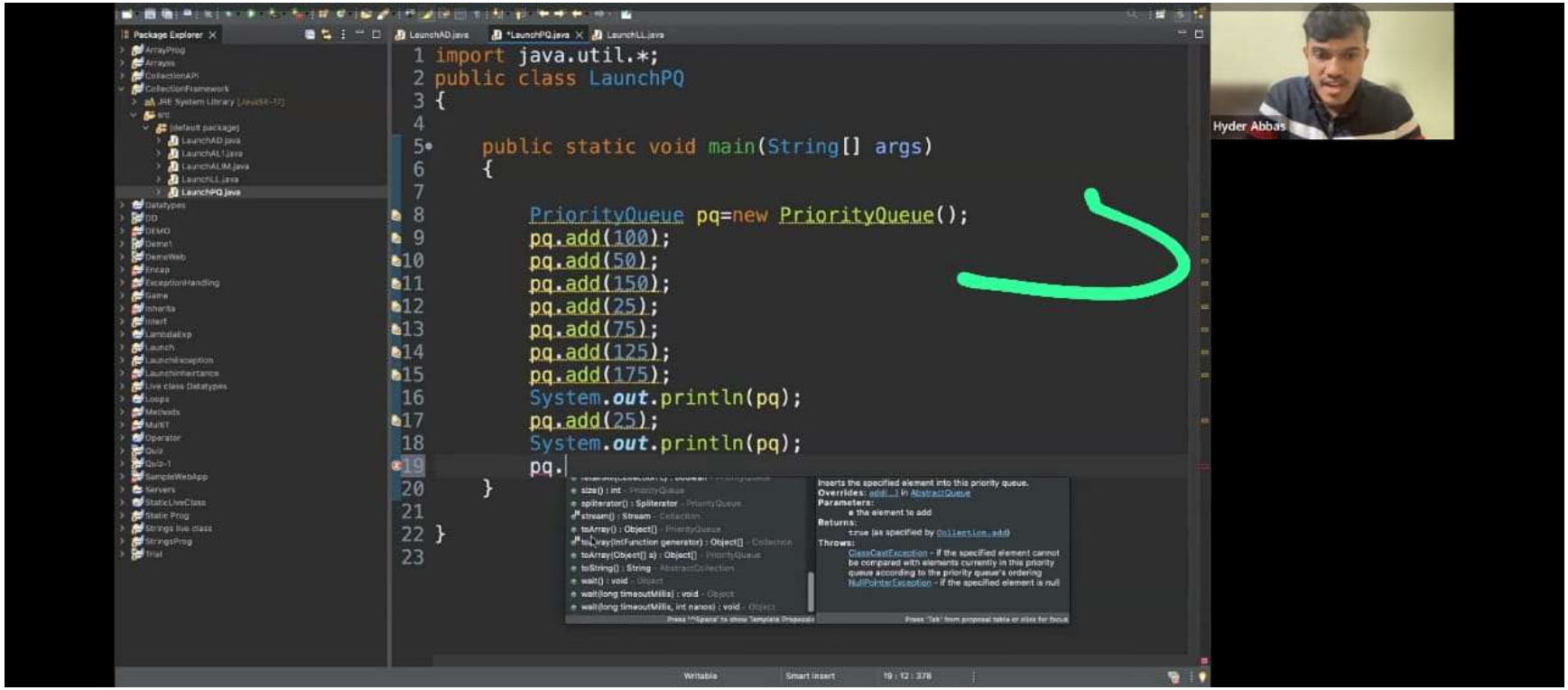

Package Explorer
LaunchAD.java
*LaunchPQ.java
1 import java.util.*;
2 public class LaunchPQ
3 {
4
5 public static void main(String[] args)
6 {
7
8 PriorityQueue pq=new PriorityQueue();
9 pq.add
10 }
11
12 }
13
add(Object e) : boolean - PriorityQueue
addAll(Collection c) : boolean - AbstractQueue
Inserts the specified element into this priority queue.
Overrides: add(...) in AbstractQueue
Parameters:
e the element to add
Returns:
true (as specified by Collection.add)
Throws:
ClassCastException - if the specified element cannot be compared with elements currently in this priority queue according to the priority queue's ordering
NullPointerException - if the specified element is null
Writable
Smart insert
9 : 15 : 153
Hyder Abbas

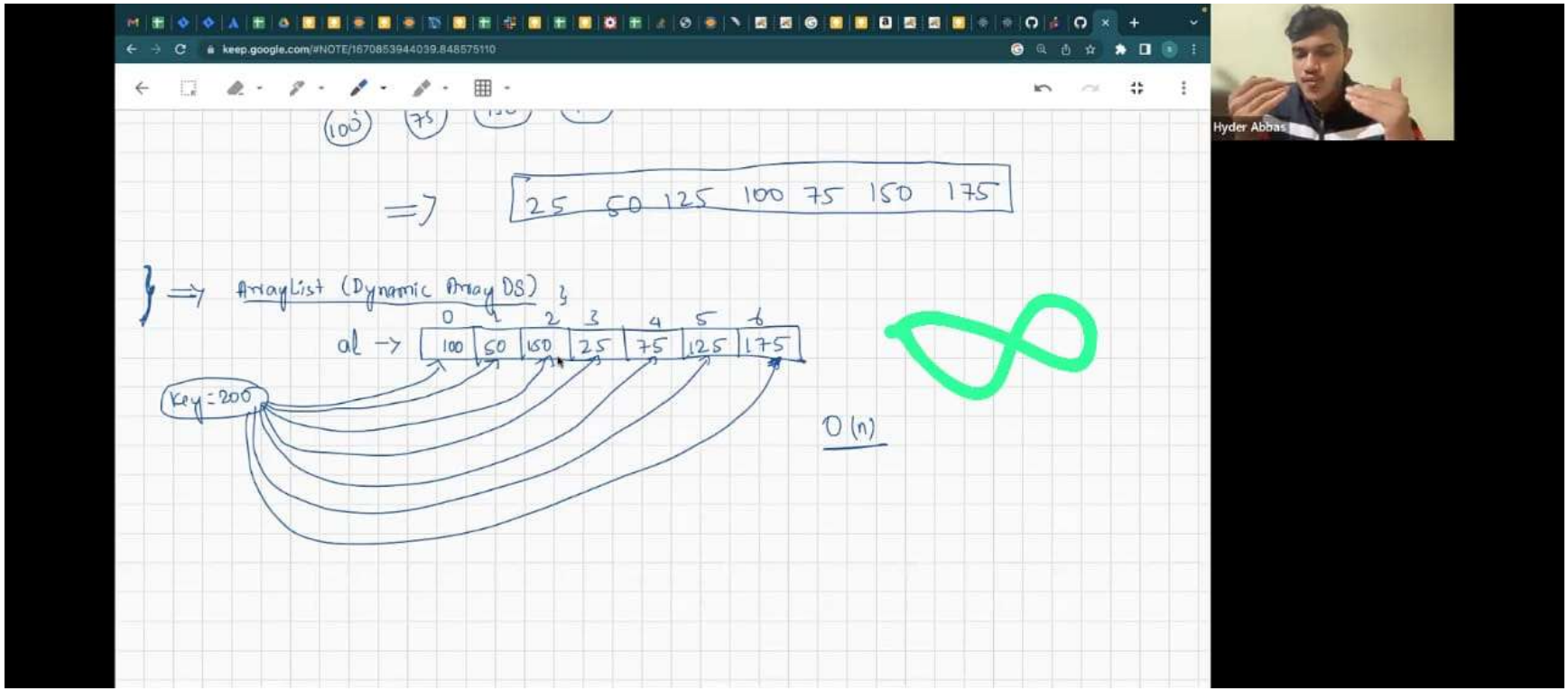

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
1 import java.util.*;
2 public class LaunchPQ
3 {
4
5 public static void main(String[] args)
6 {
7
8 PriorityQueue pq=new PriorityQueue();
9 pq.add(100);
10 pq.add(50);
11 pq.add(150);
12 pq.add(25);
13 pq.add(75);
14 pq.add(125);
15 pq.add(175);
16 System.out.println(pq);
17 }
18
19 }
20
Hyder Abbas
Writable
Smart Insert

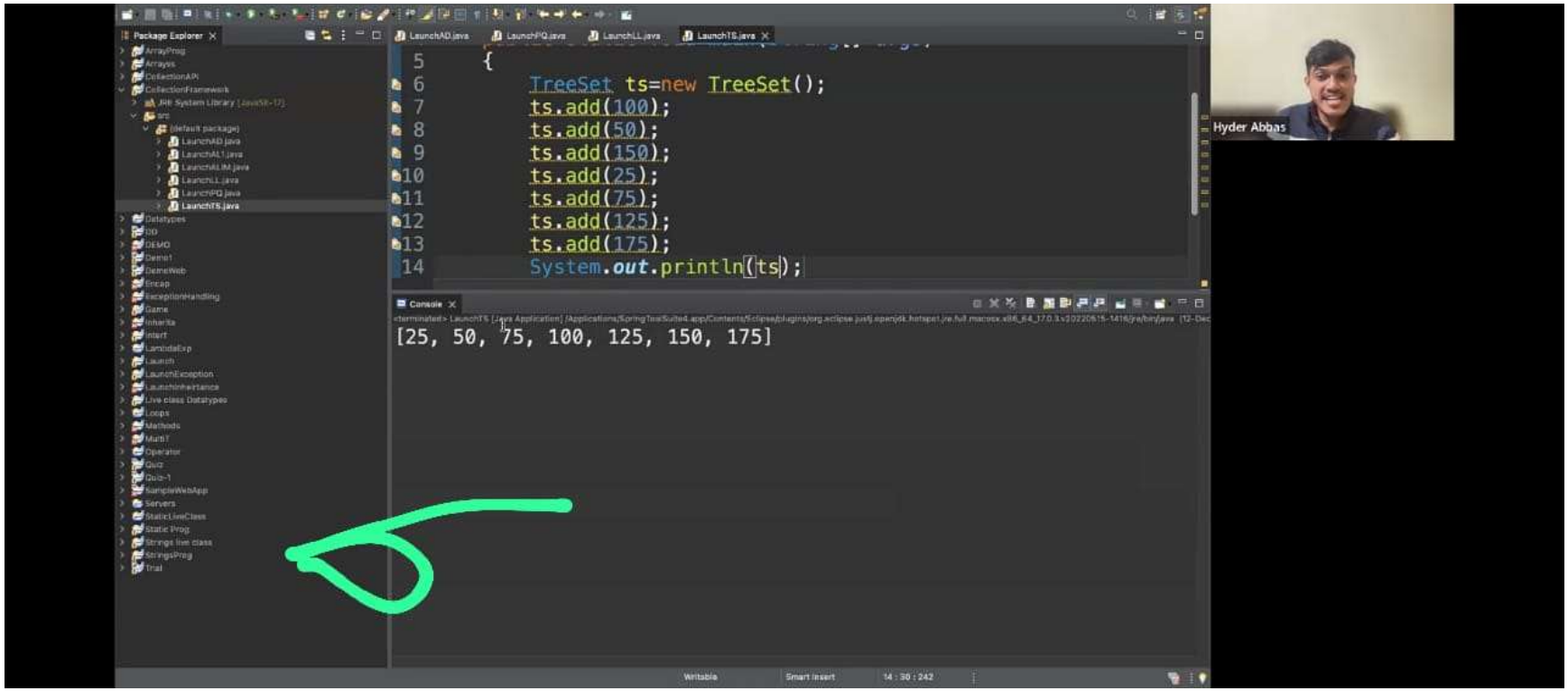

keep.google.com/#NOTE/1670853944039.848575110
Linked Hashset
PriorityQueue } => (min-heap Data Structure) }
Queue (I)
=> min-heap }
I/P
=> 100 50 150 25 75 125 175
O/P
=> 25 50 125 100 75 1
2
8
1
7
5
1
2
Hyder Abbas

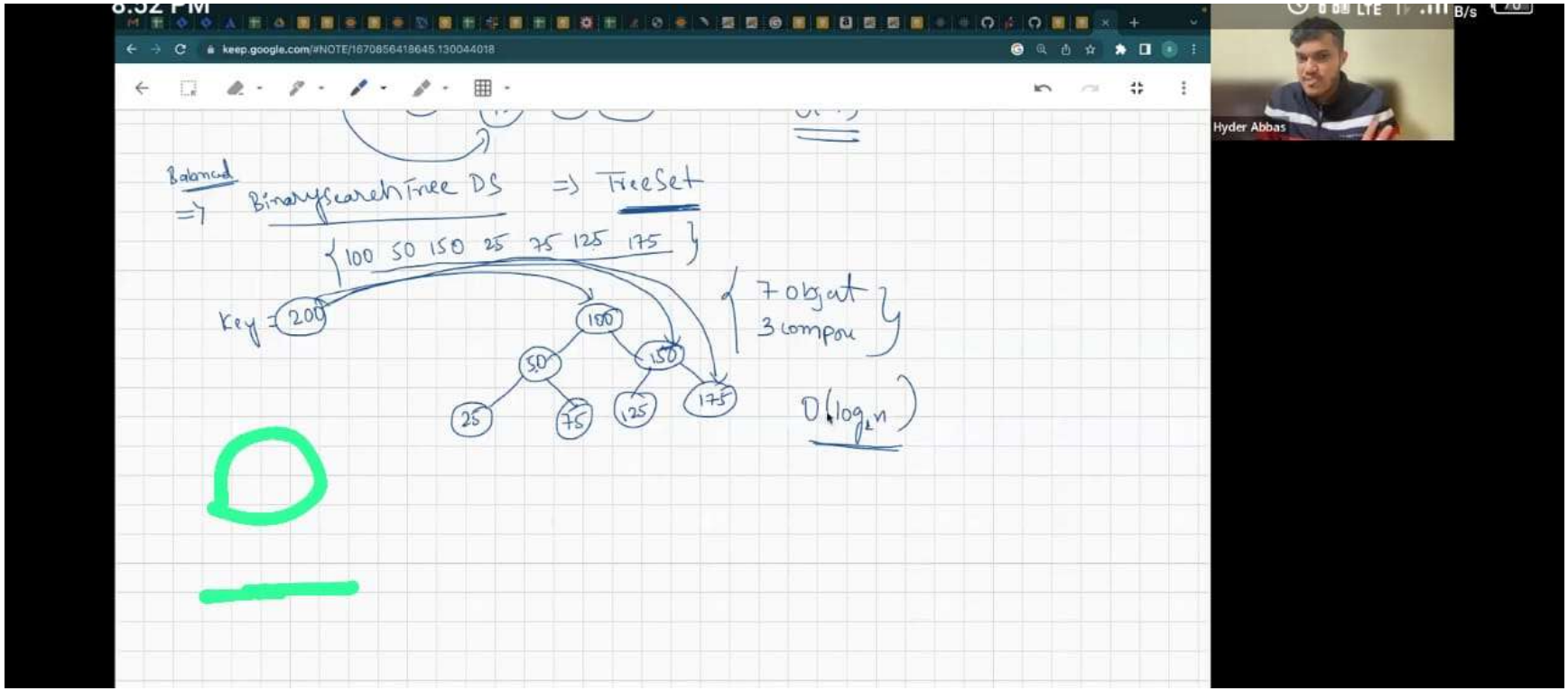

keep.google.com/#NOTE/1670853944039.848575110
Hyder Abbas

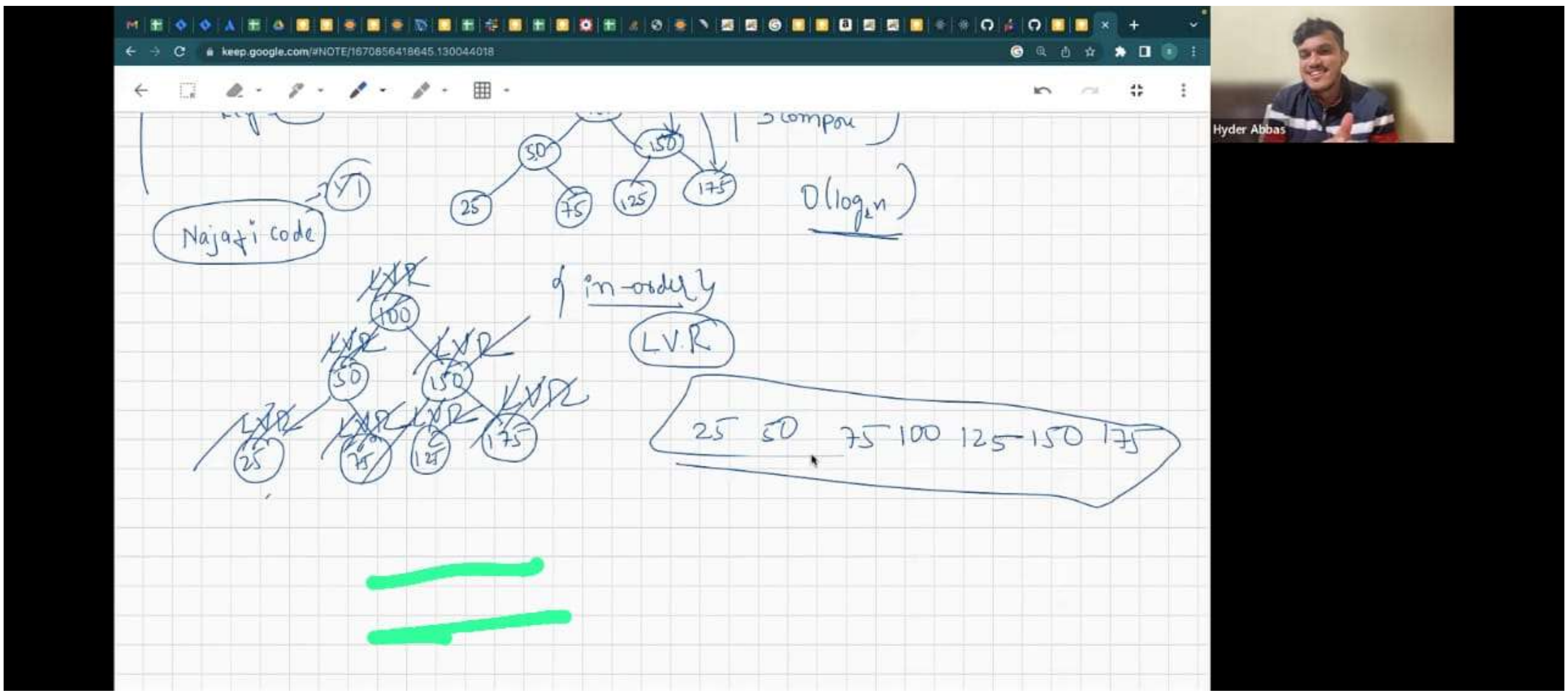

keep.google.com/#NOTE/1670853944039.848575110
Hyder Abbas
25
50
150
100
75
125
<=>
50
125
100
75
150
175
=>
25
50
125
100
75
150
175
0th level
first level
second level
=> 25 50 125 100 75 150 175

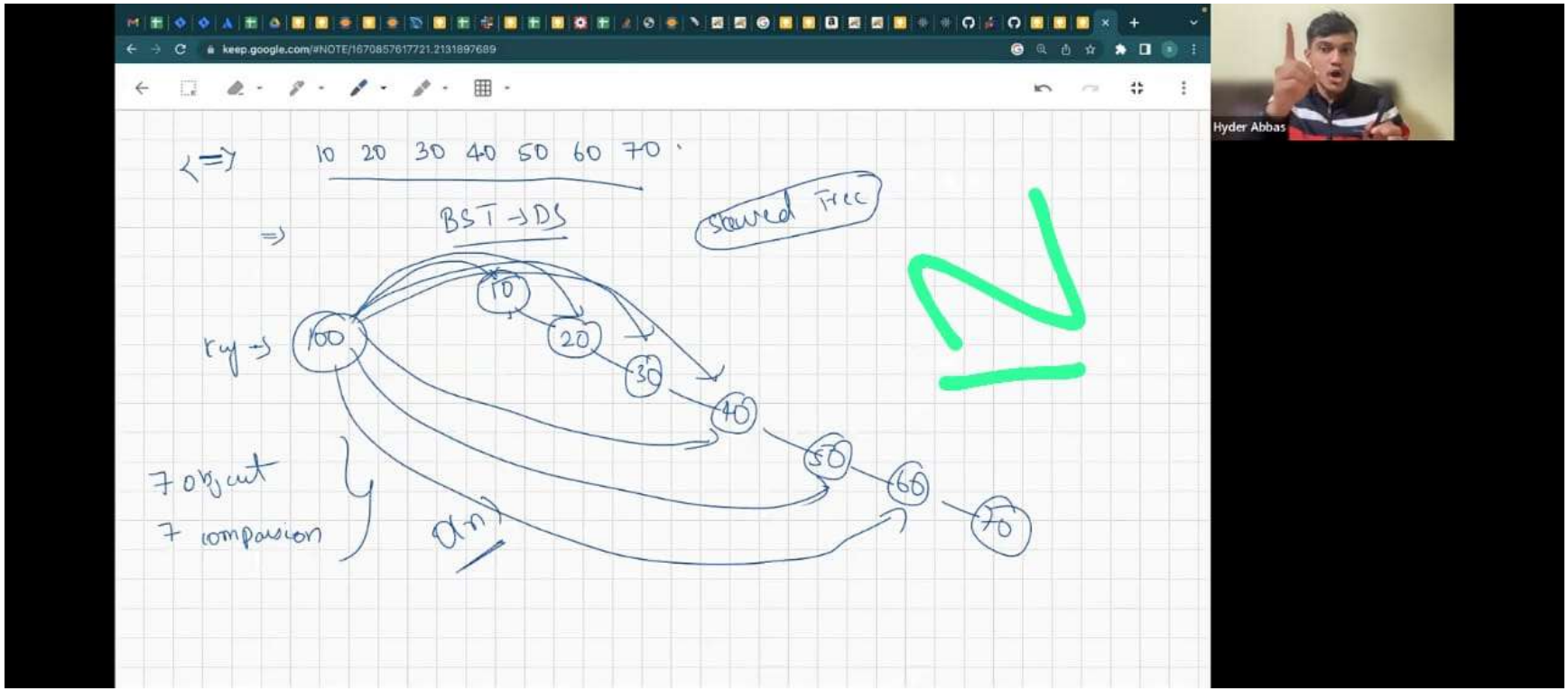

Package Explorer
LaunchAD.java
*LaunchPQ.java
LaunchLL.java
1 import java.util.*;
2 public class LaunchPQ
3 {
4
5 public static void main(String[] args)
6 {
7
8 PriorityQueue pq=new PriorityQueue();
9 pq.add(100);
10 pq.add(50);
11 pq.add(150);
12 pq.add(25);
13 pq.add(75);
14 pq.add(125);
15 pq.add(175);
16 System.out.println(pq);
17 pq.add(25);
18 System.out.println(pq);
19 pq.
20 }
21
22 }
23
size() : int - PriorityQueue
spliterator() : Spliterator - PriorityQueue
stream() : Stream - Collection
toArray() : Object[] - PriorityQueue
toArray(IntFunction generator) : Object[] - Collection
toArray(Object[] a) : Object[] - PriorityQueue
toString() : String - AbstractCollection
wait() : void - Object
wait(long timeoutMillis) : void - Object
wait(long timeoutMillis, int nanos) : void - Object
Inserts the specified element into this priority queue.
Overrides: add(...) in AbstractQueue
Parameters:
e the element to add
Returns:
true (as specified by Collection.add)
Throws:
ClassCastException - if the specified element cannot be compared with elements currently in this priority queue according to the priority queue's ordering
NullPointerException - if the specified element is null
Writable
Smart insert
19 : 12 : 378
Hyder Abbas

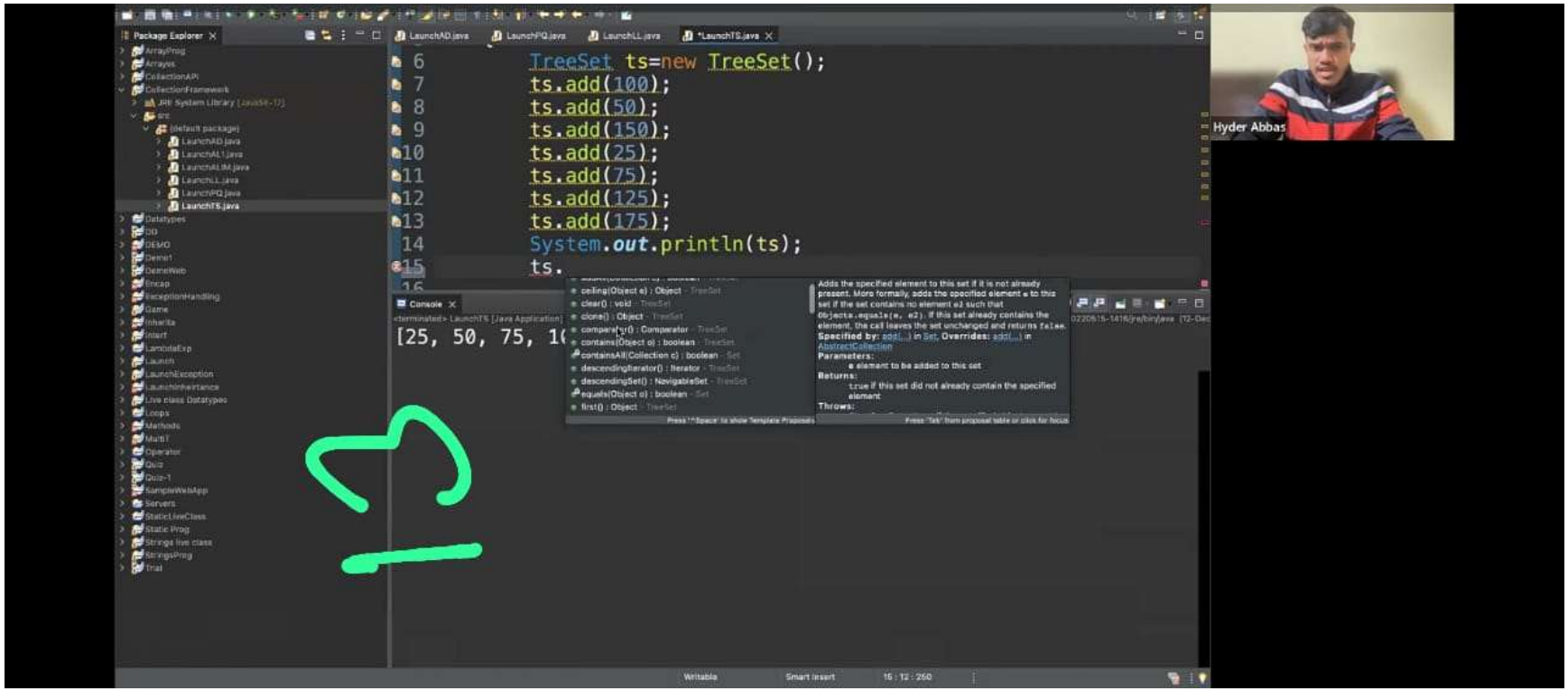

keep.google.com/#NOTE/1670853944039.848575110
100
75
=> 25 50 125 100 75 150 175
=> ArrayList (Dynamic Array DS)
al ->
0 1 2 3 4 5 6
100 50 150 25 75 125 175
key = 200
O(n)
Hyder Abbas

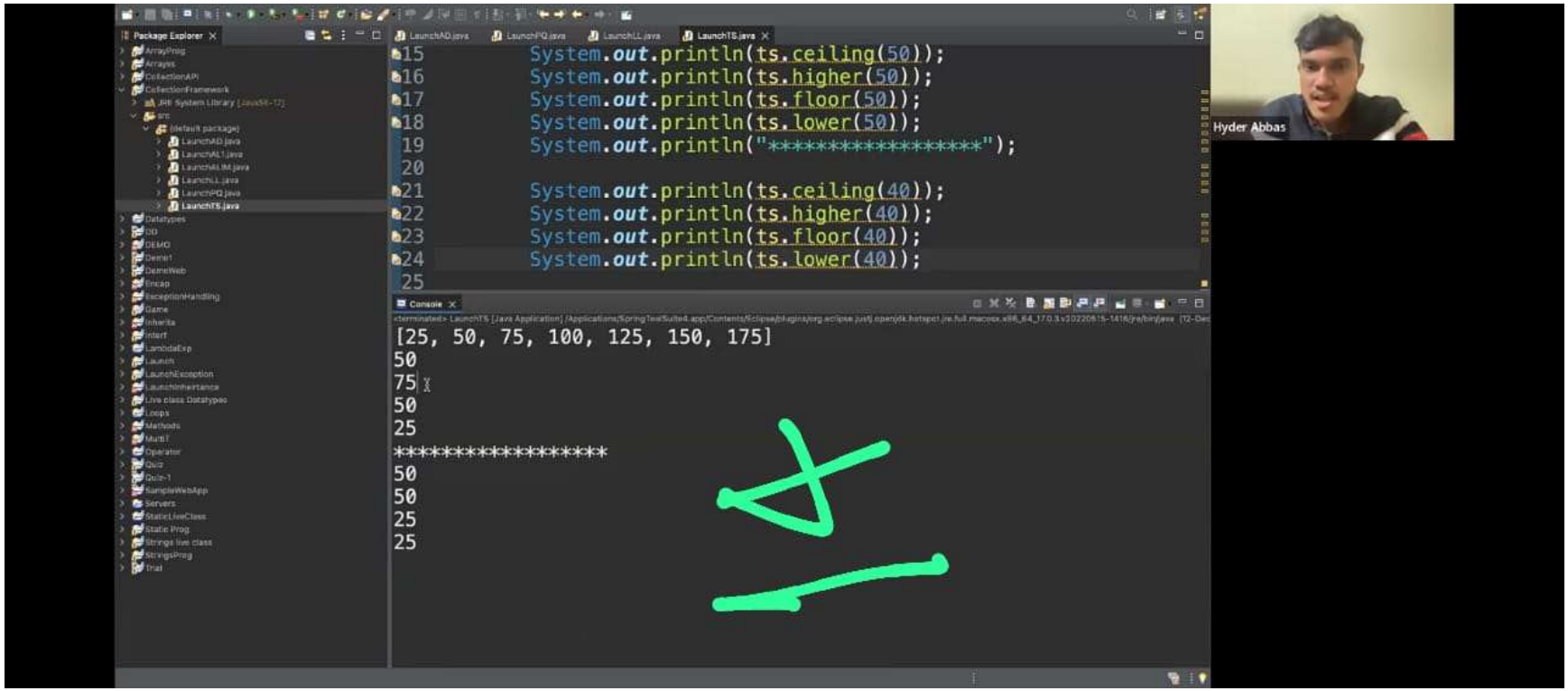

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
{
TreeSet ts=new TreeSet();
ts.add(100);
ts.add(50);
ts.add(150);
ts.add(25);
ts.add(75);
ts.add(125);
ts.add(175);
System.out.println(ts);
Console
[25, 50, 75, 100, 125, 150, 175]
Writable
Smart insert
14 : 30 : 242
Hyder Abbas

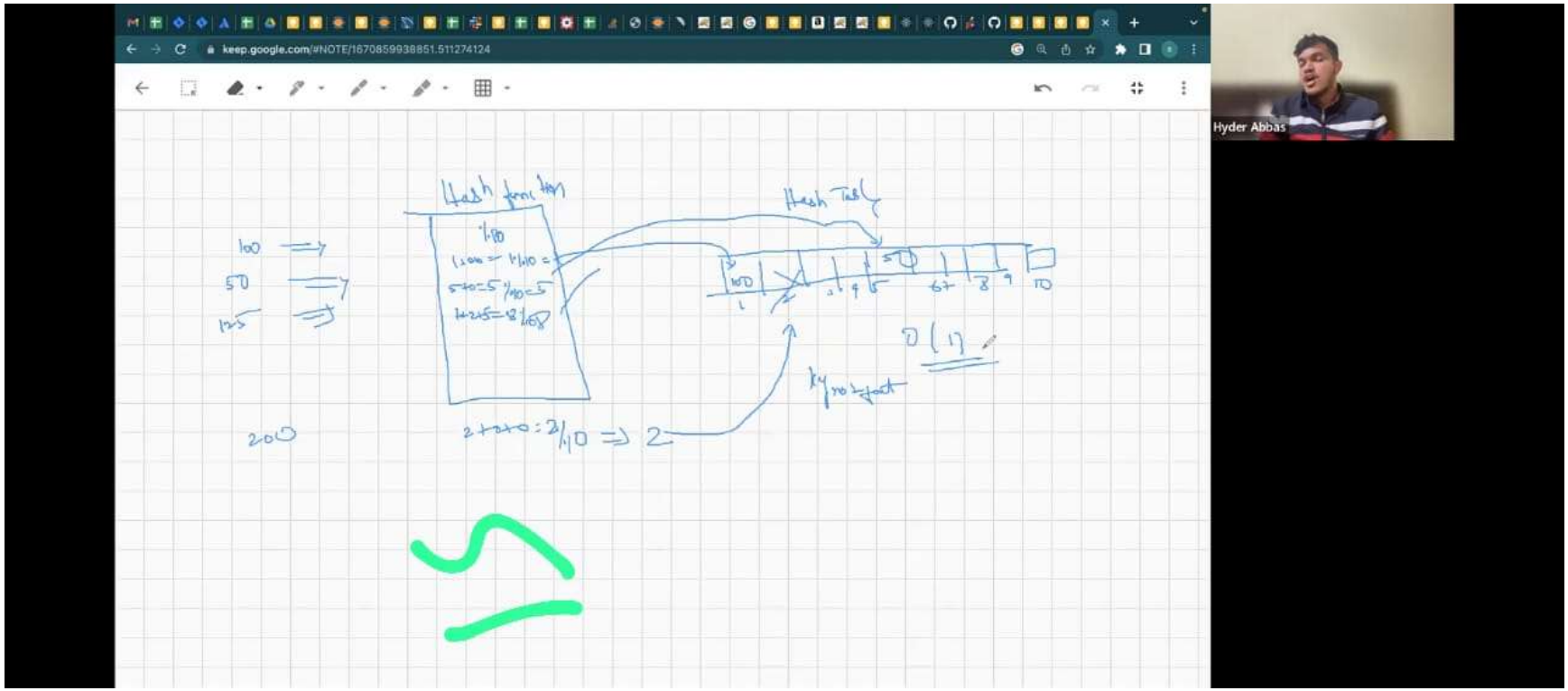

keep.google.com/#NOTE/1670856418645.130044018
Balanced
=> BinarySearchTree DS => TreeSet
{ 100 50 150 25 75 125 175 }
Key = 200
100
50
150
25
75
125
175
{ 7 object
3 compar }
O(log2n)
Hyder Abbas

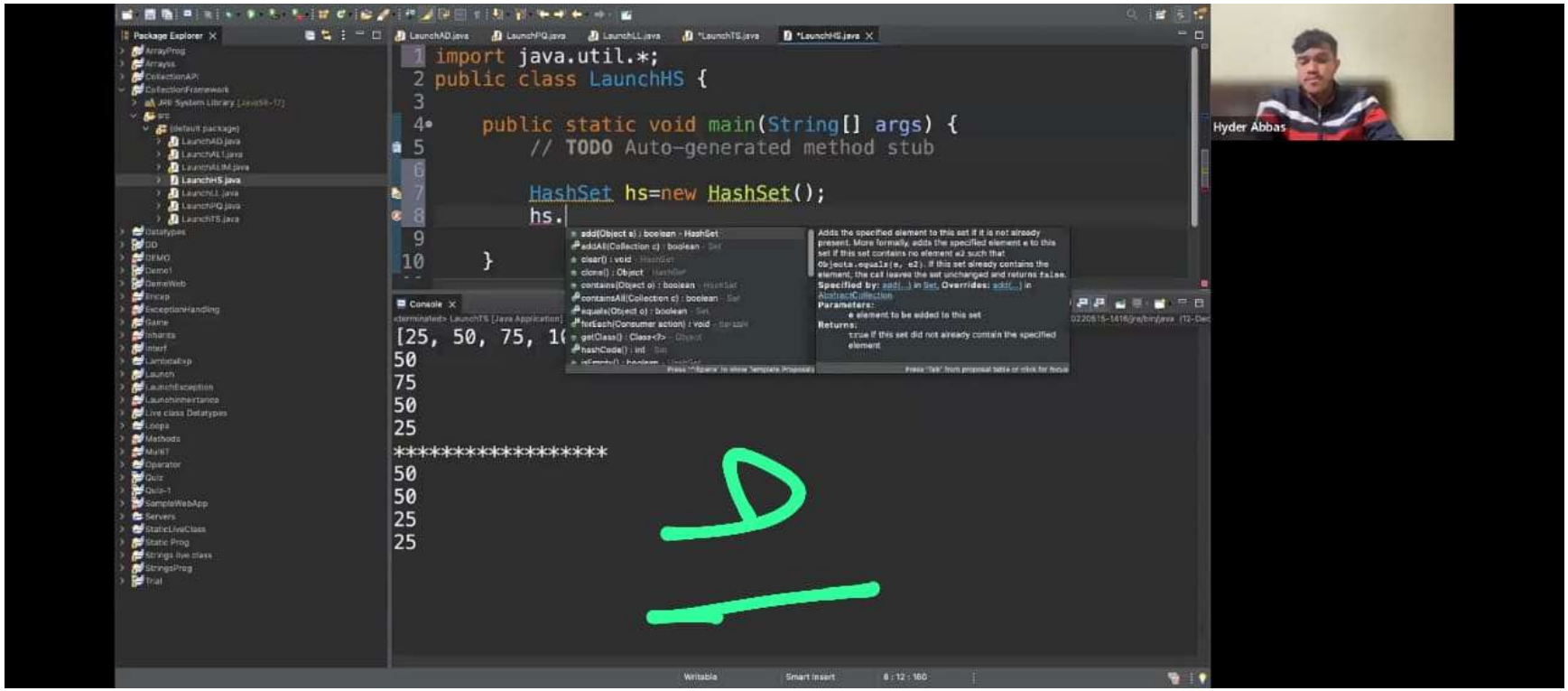

keep.google.com/#NOTE/1670856418645.130044018
Hyder Abbas
3 compar
O(log₂n)
Najafi code
YT
in-order
L.V.R
25 50 75 100 125 150 175

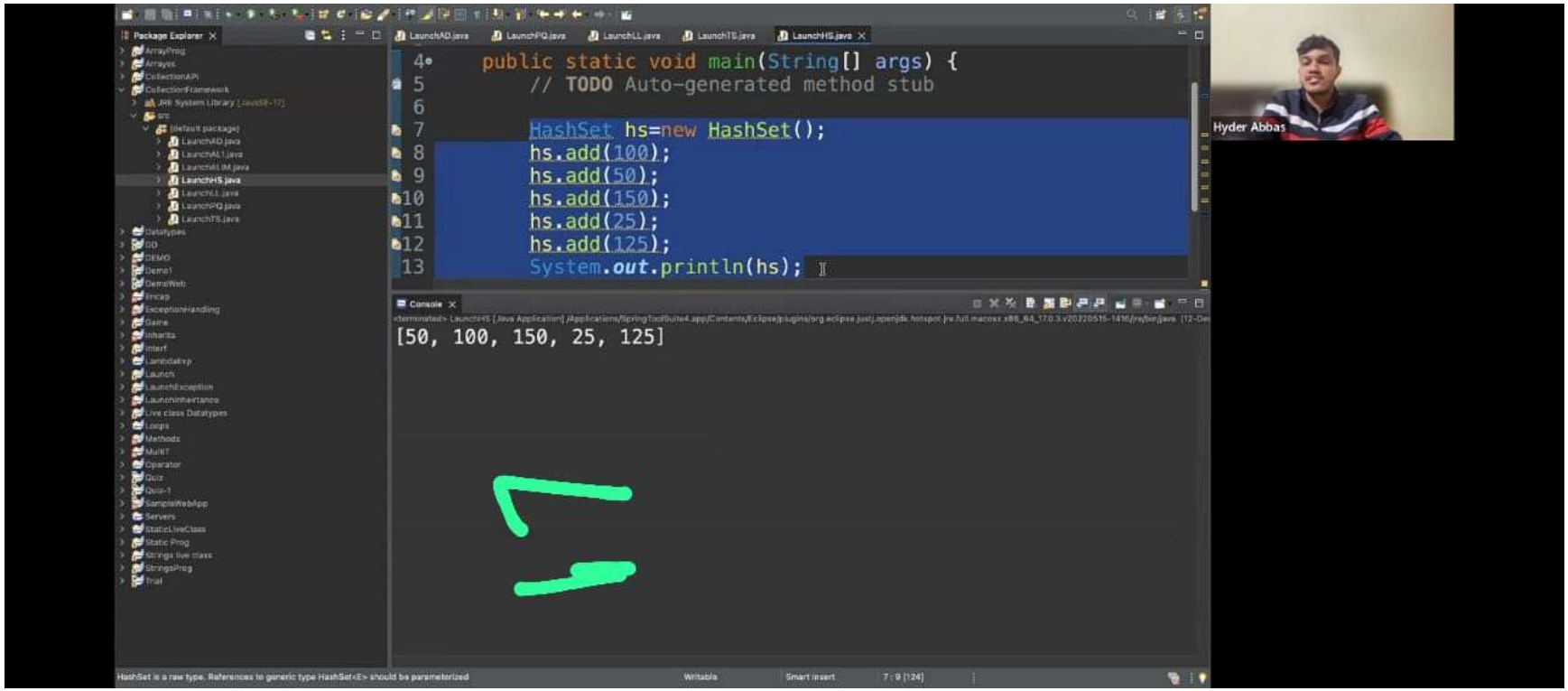

keep.google.com/#NOTE/1670857617721.2131897689
10 20 30 40 50 60 70
BST → DS
Stored Tree
try →
100
10
20
30
40
50
60
70
7 object
7 comparison
O(n)
Hyder Abbas

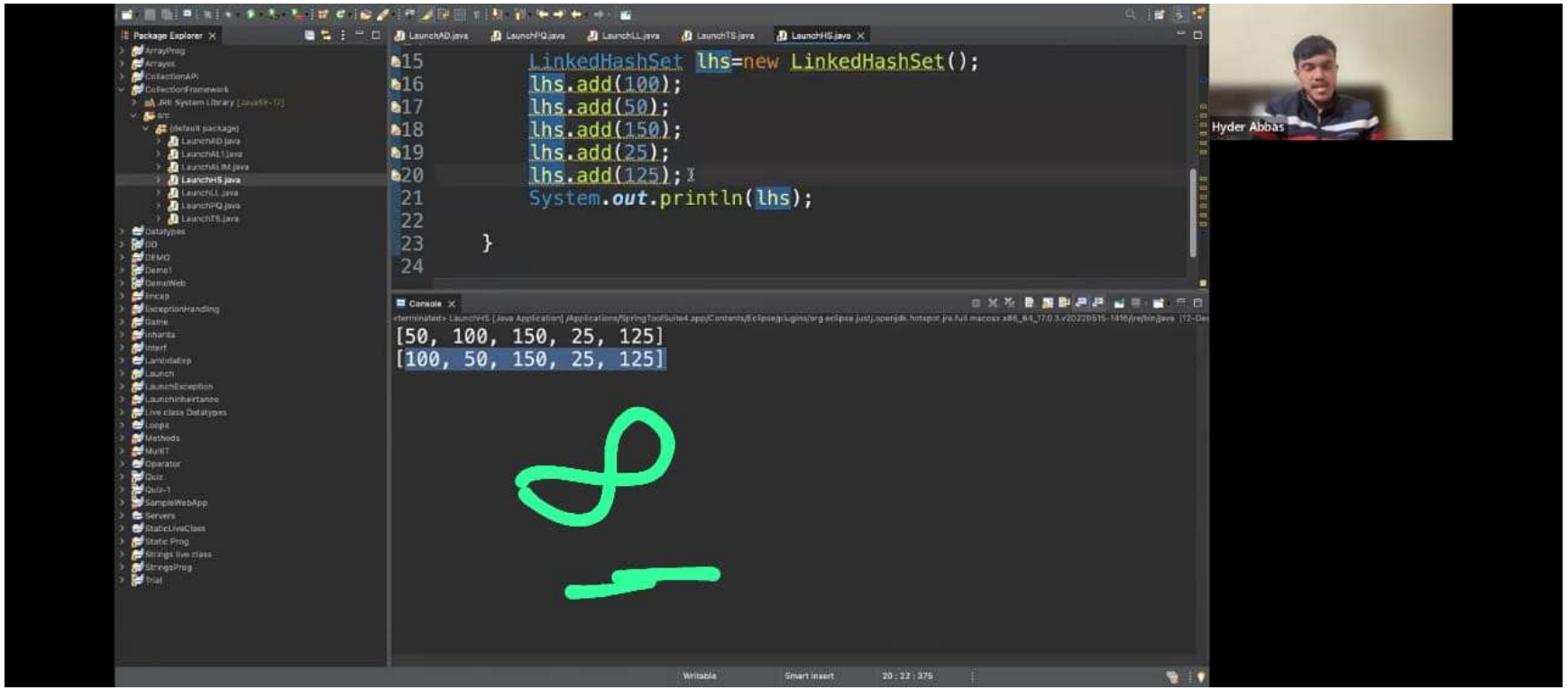

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
*LaunchTS.java
TreeSet ts=new TreeSet();
ts.add(100);
ts.add(50);
ts.add(150);
ts.add(25);
ts.add(75);
ts.add(125);
ts.add(175);
System.out.println(ts);
ts.
Console
[25, 50, 75, 1
Hyder Abbas

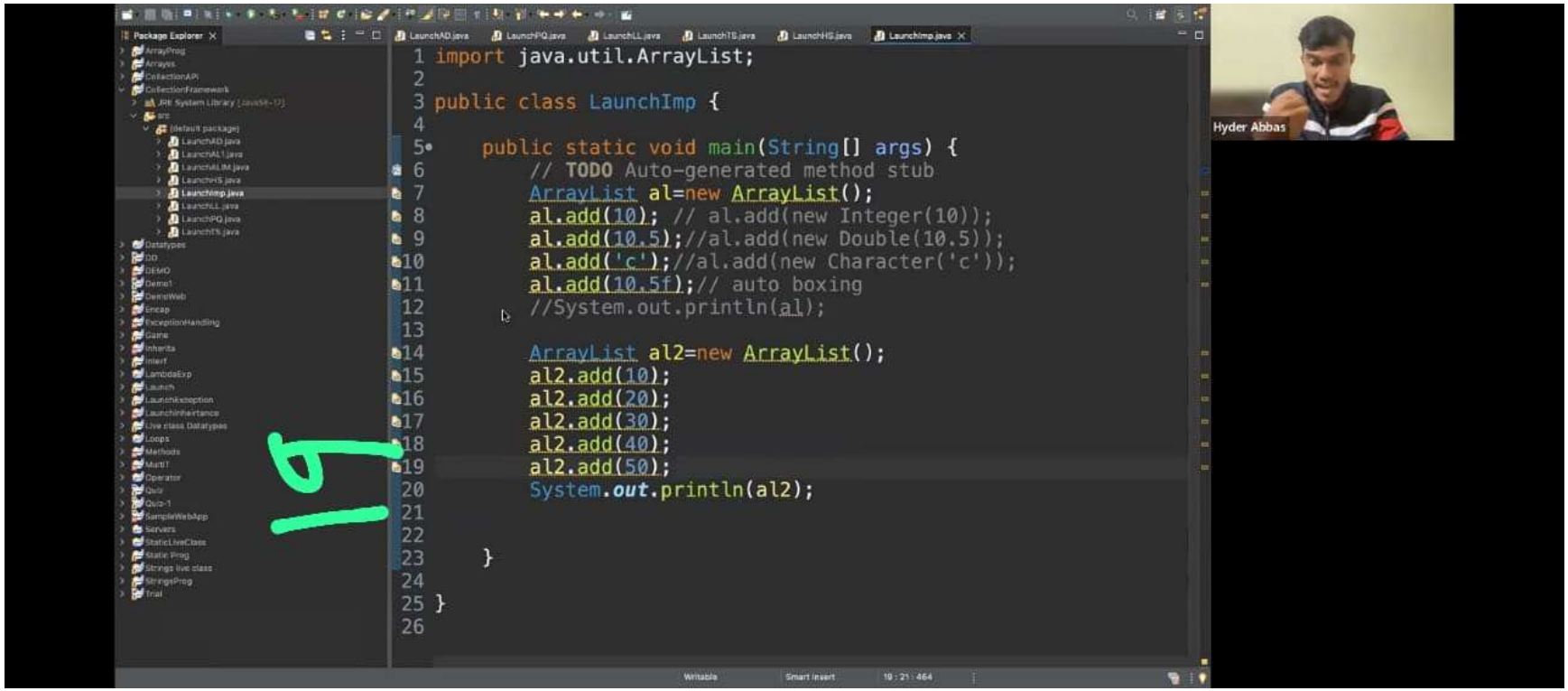

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
System.out.println(ts.ceiling(50));
System.out.println(ts.higher(50));
System.out.println(ts.floor(50));
System.out.println(ts.lower(50));
System.out.println("**********");
System.out.println(ts.ceiling(40));
System.out.println(ts.higher(40));
System.out.println(ts.floor(40));
System.out.println(ts.lower(40));
Console
[25, 50, 75, 100, 125, 150, 175]
50
75
50
25
****
50
50
25
25
Hyder Abbas

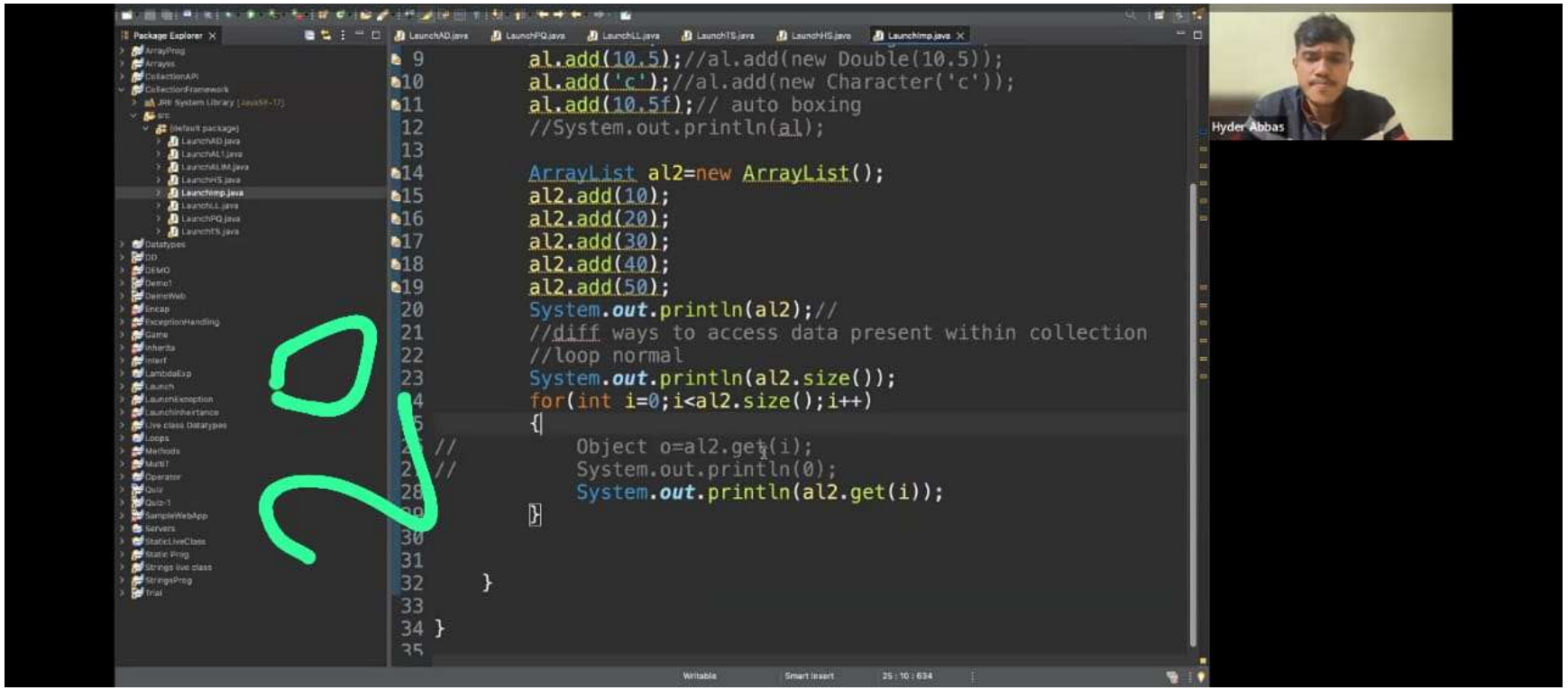

keep.google.com/#NOTE/1670859938851.511274124
Hash function
Hash Table
100
50
125
200
O(1)
Key not found
Hyder Abbas


Hyder Abbas
import java.util.*;
public class LaunchHS {
public static void main(String[] args) {
// TODO Auto-generated method stub
HashSet hs=new HashSet();
hs.
}
[25, 50, 75, 1
50
75
50
25
****
50
50
25
25

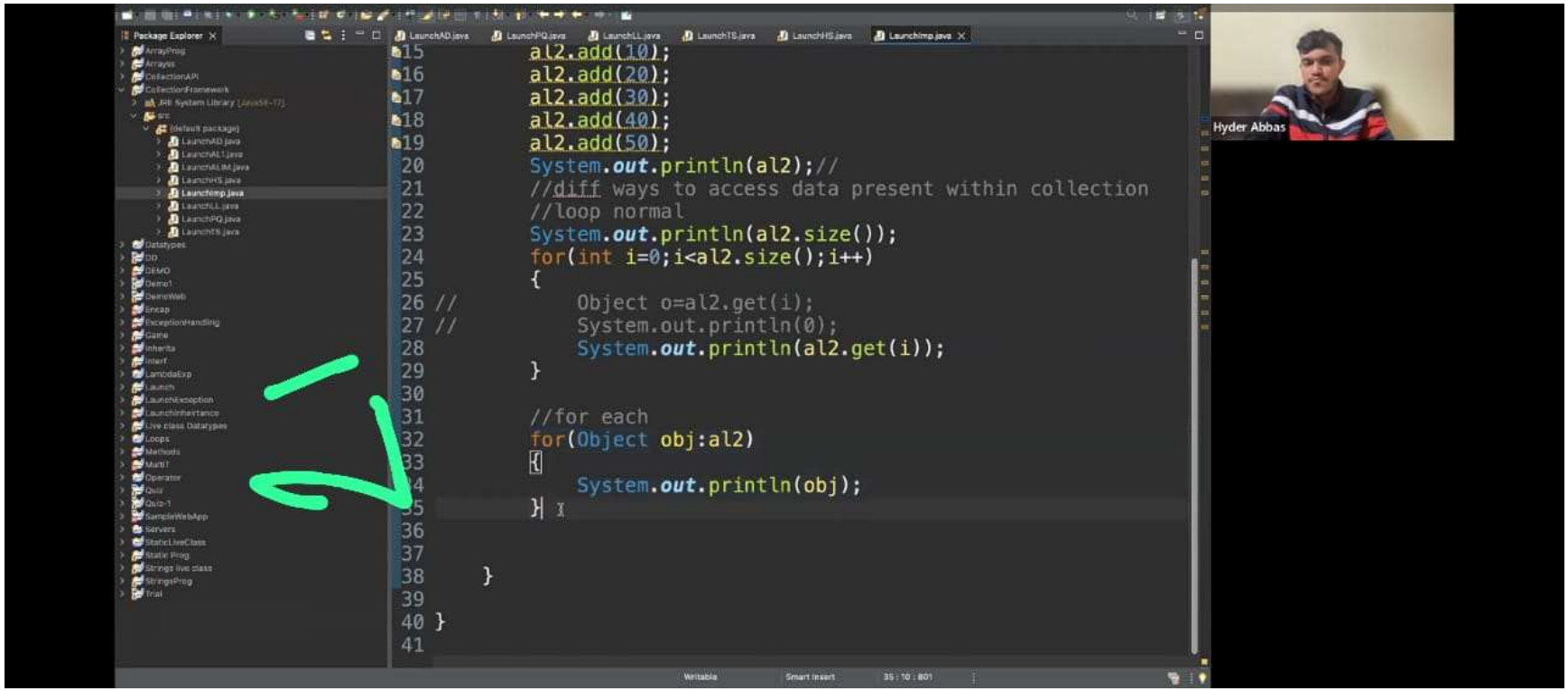

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
public static void main(String[] args) {
// TODO Auto-generated method stub
HashSet hs=new HashSet();
hs.add(100);
hs.add(50);
hs.add(150);
hs.add(25);
hs.add(125);
System.out.println(hs);
Console
[50, 100, 150, 25, 125]
Hyder Abbas
HashSet is a raw type. References to generic type HashSet<E> should be parameterized
Writable
Smart insert
7 : 9 [124]

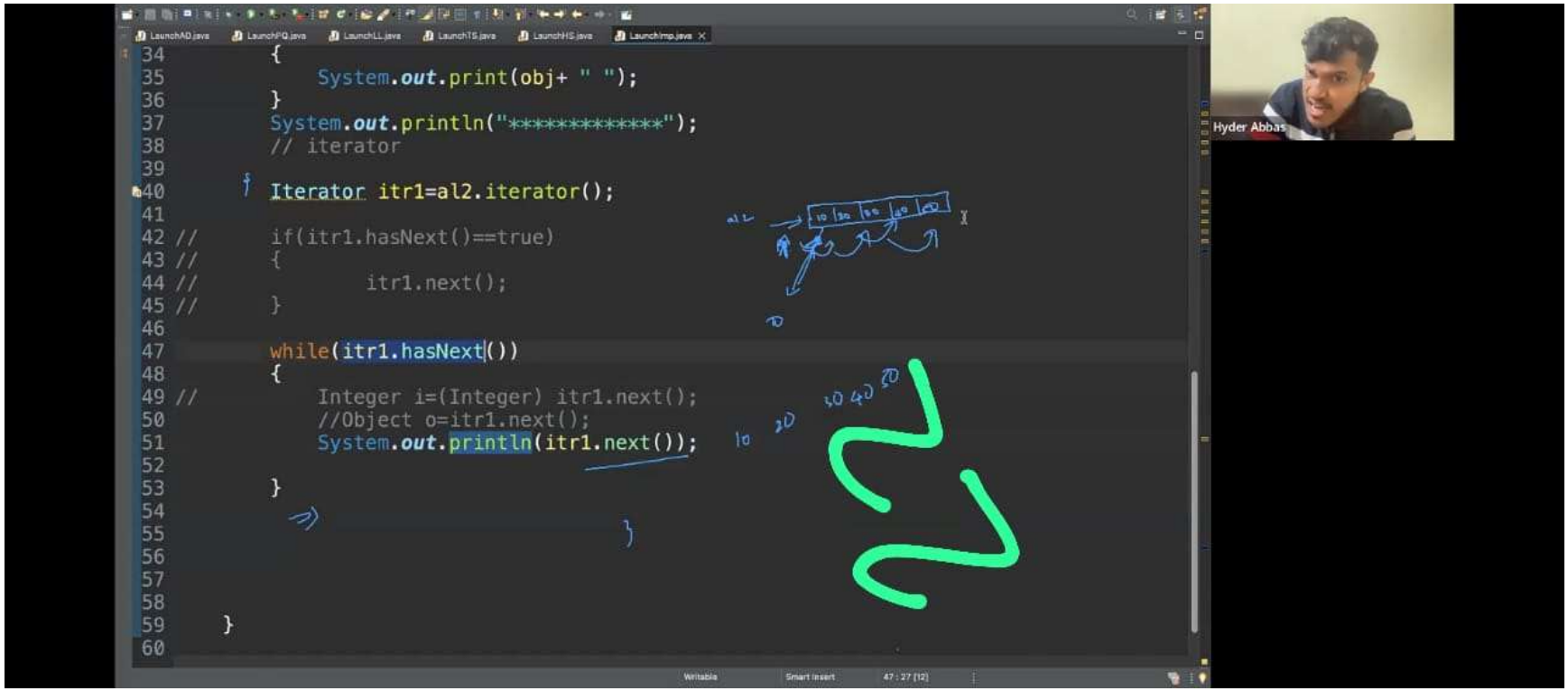

Package Explorer
LaunchAD.java
LaunchFIQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LinkedHashSet lhs=new LinkedHashSet();
lhs.add(100);
lhs.add(50);
lhs.add(150);
lhs.add(25);
lhs.add(125);
System.out.println(lhs);
}
Console
[50, 100, 150, 25, 125]
[100, 50, 150, 25, 125]
Hyder Abbas
Writable
Smart Insert
20 : 22 : 375

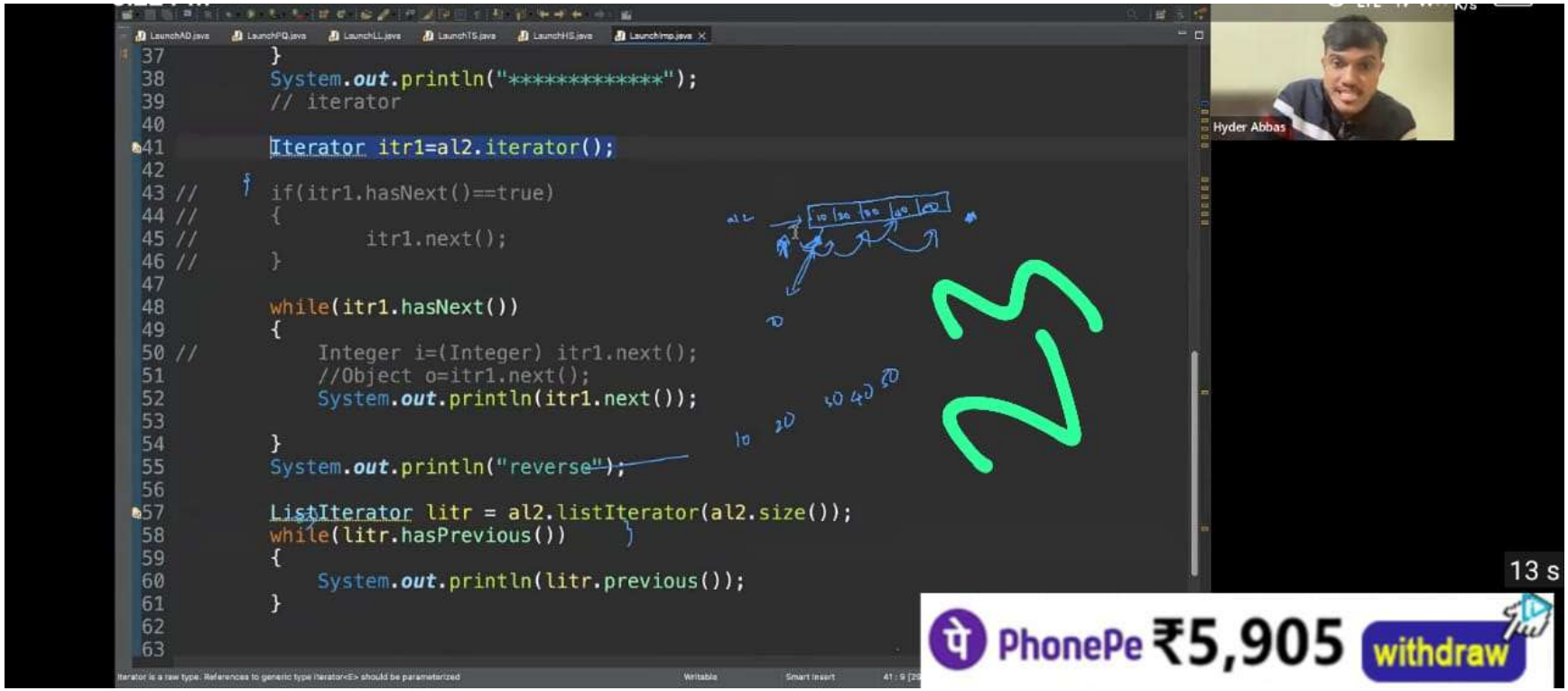

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LaunchImp.java
1 import java.util.ArrayList;
2
3 public class LaunchImp {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7 ArrayList al=new ArrayList();
8 al.add(10); // al.add(new Integer(10));
9 al.add(10.5);//al.add(new Double(10.5));
10 al.add('c');//al.add(new Character('c'));
11 al.add(10.5f);// auto boxing
12 //System.out.println(al);
13
14 ArrayList al2=new ArrayList();
15 al2.add(10);
16 al2.add(20);
17 al2.add(30);
18 al2.add(40);
19 al2.add(50);
20 System.out.println(al2);
21
22
23 }
24
25 }
26
19
Writable
Smart Insert
19 : 21 : 464
Hyder Abbas

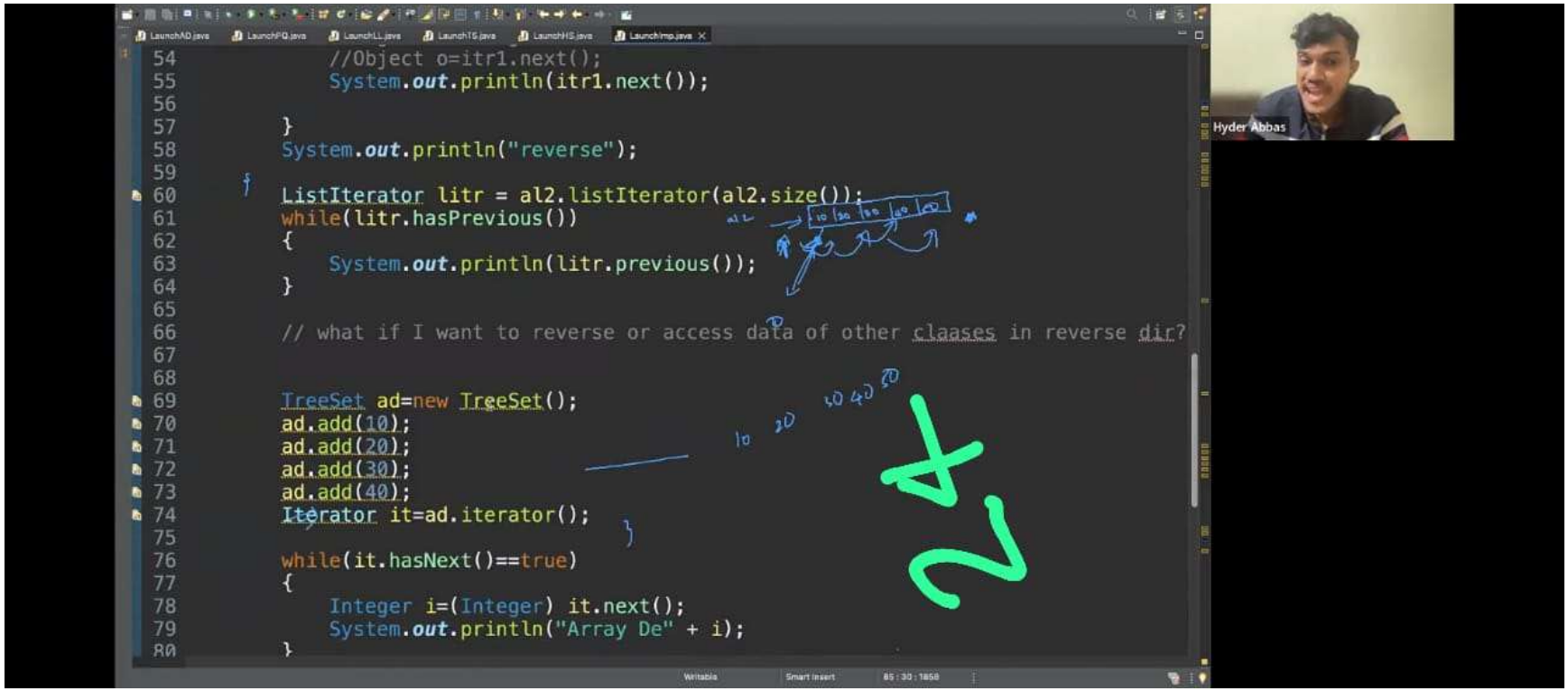


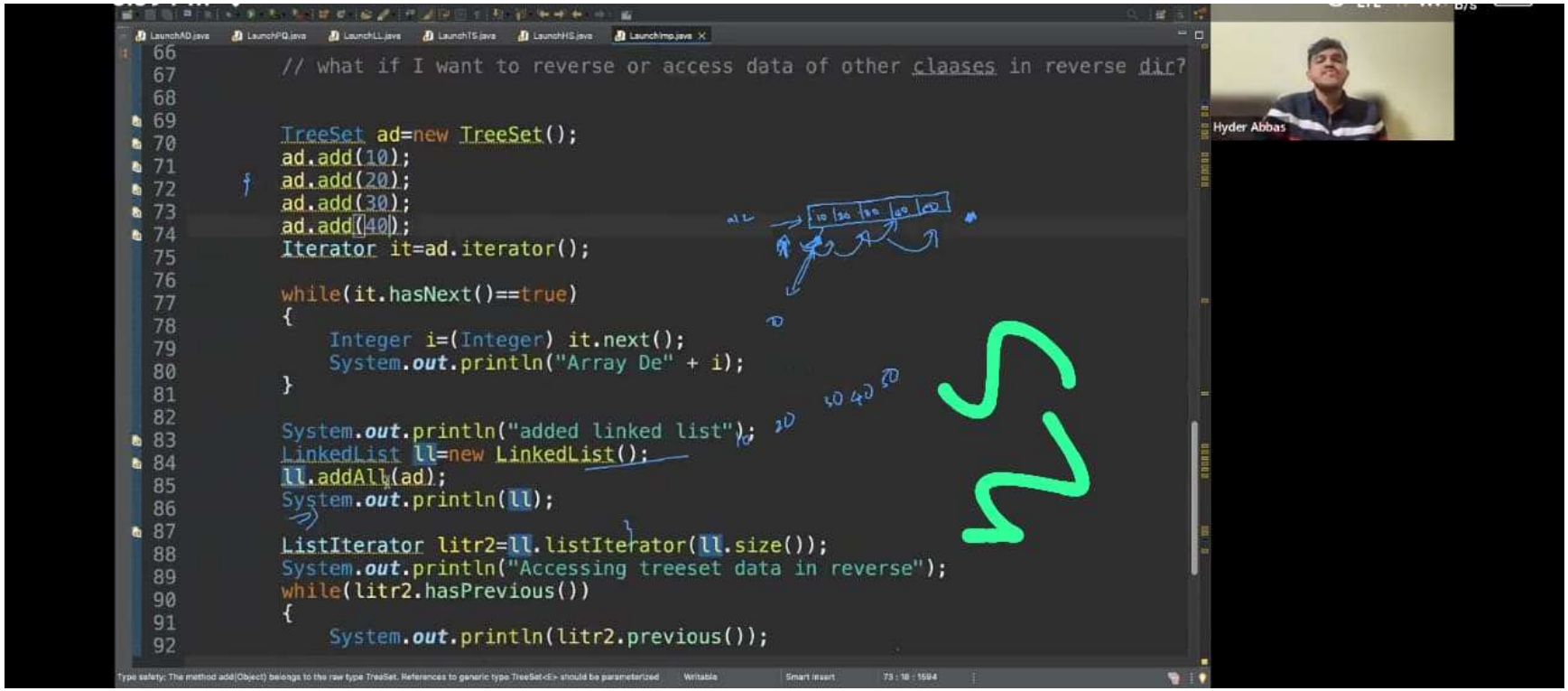

```
        al.add(10.5);//al.add(new Double(10.5));
        al.add('c');//al.add(new Character('c'));
        al.add(10.5f);// auto boxing
        //System.out.println(al);

        ArrayList al2=new ArrayList();
        al2.add(10);
        al2.add(20);
        al2.add(30);
        al2.add(40);
        al2.add(50);
        System.out.println(al2);//
        //diff ways to access data present within collection
        //loop normal
        System.out.println(al2.size());
        for(int i=0;i<al2.size();i++)
        {
//          Object o=al2.get(i);
//          System.out.println(0);
            System.out.println(al2.get(i));
        }


    }

}
```

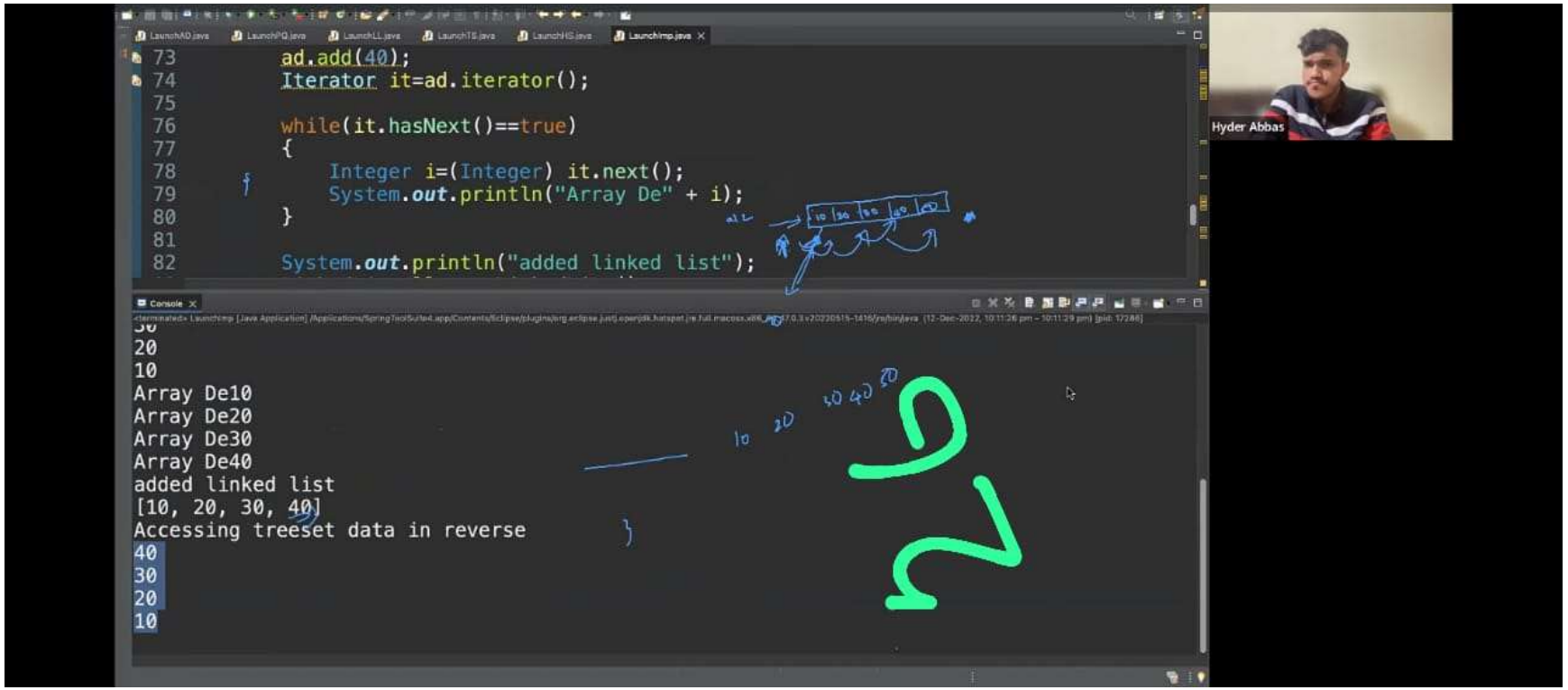

Package Explorer
LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LaunchImp.java
al2.add(10);
al2.add(20);
al2.add(30);
al2.add(40);
al2.add(50);
System.out.println(al2);//
//diff ways to access data present within collection
//loop normal
System.out.println(al2.size());
for(int i=0;i<al2.size();i++)
{
// Object o=al2.get(i);
// System.out.println(0);
System.out.println(al2.get(i));
}
//for each
for(Object obj:al2)
{
System.out.println(obj);
}
}
}
Writable
Smart Insert
35 : 10 : 801
Hyder Abbas


LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LaunchImp.java
{
System.out.print(obj+ " ");
}
System.out.println("**********");
// iterator
Iterator itr1=al2.iterator();
// if(itr1.hasNext()==true)
// {
// itr1.next();
// }
while(itr1.hasNext())
{
// Integer i=(Integer) itr1.next();
//Object o=itr1.next();
System.out.println(itr1.next());
}
}
Hyder Abbas
Writable
Smart Insert
47 : 27 [12]


LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LaunchImp.java
}
System.out.println("**********");
// iterator
Iterator itr1=al2.iterator();
// if(itr1.hasNext()==true)
// {
// itr1.next();
// }
while(itr1.hasNext())
{
// Integer i=(Integer) itr1.next();
//Object o=itr1.next();
System.out.println(itr1.next());
}
System.out.println("reverse");
ListIterator litr = al2.listIterator(al2.size());
while(litr.hasPrevious())
{
System.out.println(litr.previous());
}
Hyder Abbas
13 s
PhonePe ₹5,905
withdraw


LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LaunchImp.java
//Object o=itr1.next();
System.out.println(itr1.next());
}
System.out.println("reverse");
ListIterator litr = al2.listIterator(al2.size());
while(litr.hasPrevious())
{
System.out.println(litr.previous());
}
// what if I want to reverse or access data of other claases in reverse dir?
TreeSet ad=new TreeSet();
ad.add(10);
ad.add(20);
ad.add(30);
ad.add(40);
Iterator it=ad.iterator();
while(it.hasNext()==true)
{
Integer i=(Integer) it.next();
System.out.println("Array De" + i);
}
Writable
Smart insert
Hyder Abbas


LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
LaunchImp.java
// what if I want to reverse or access data of other claases in reverse dir?
TreeSet ad=new TreeSet();
ad.add(10);
ad.add(20);
ad.add(30);
ad.add(40);
Iterator it=ad.iterator();
while(it.hasNext()==true)
{
Integer i=(Integer) it.next();
System.out.println("Array De" + i);
}
System.out.println("added linked list");
LinkedList ll=new LinkedList();
ll.addAll(ad);
System.out.println(ll);
ListIterator litr2=ll.listIterator(ll.size());
System.out.println("Accessing treeset data in reverse");
while(litr2.hasPrevious())
{
System.out.println(litr2.previous());
Hyder Abbas


LaunchAD.java
LaunchPQ.java
LaunchLL.java
LaunchTS.java
LaunchHS.java
Launchimp.java
73 ad.add(40);
74 Iterator it=ad.iterator();
75
76 while(it.hasNext()==true)
77 {
78 Integer i=(Integer) it.next();
79 System.out.println("Array De" + i);
80 }
81
82 System.out.println("added linked list");
Console
20
10
Array De10
Array De20
Array De30
Array De40
added linked list
[10, 20, 30, 40]
Accessing treeset data in reverse
40
30
20
10
Hyder Abbas