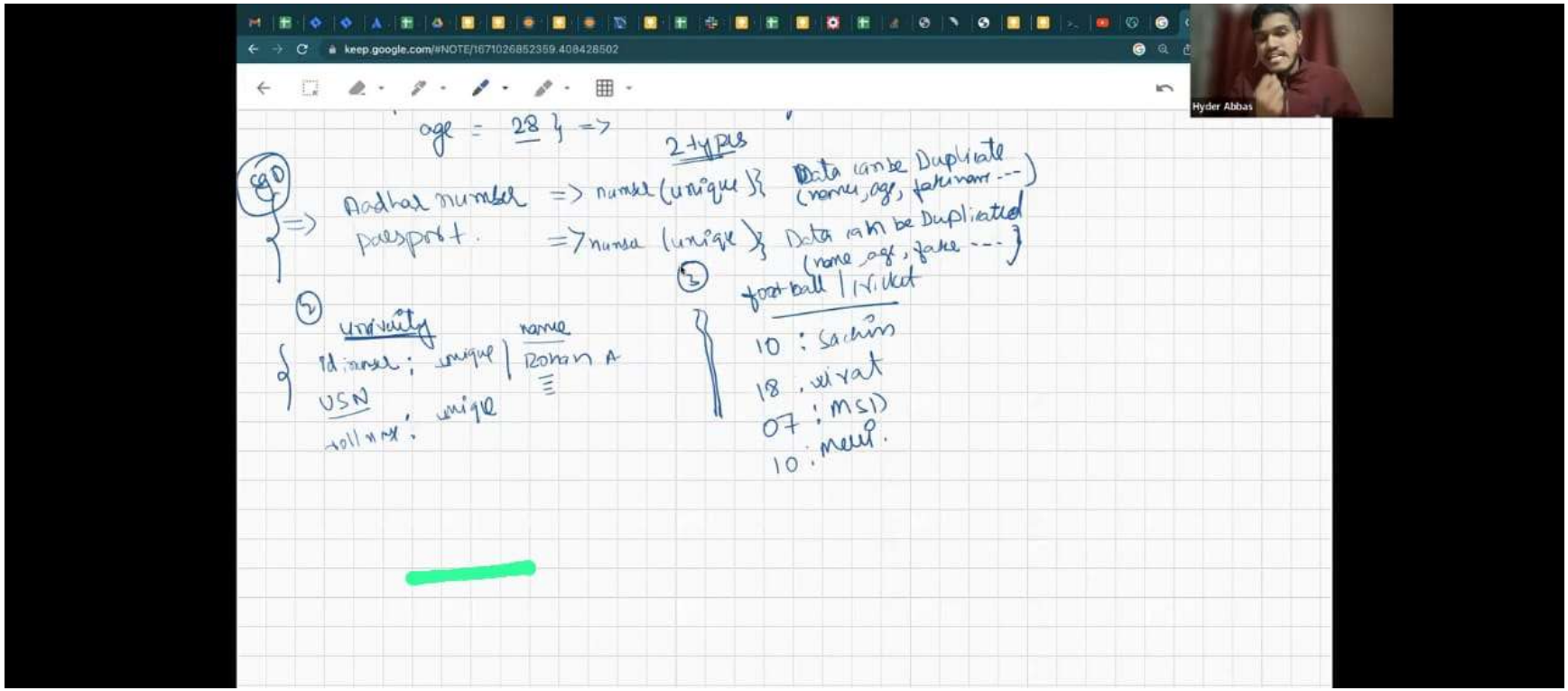
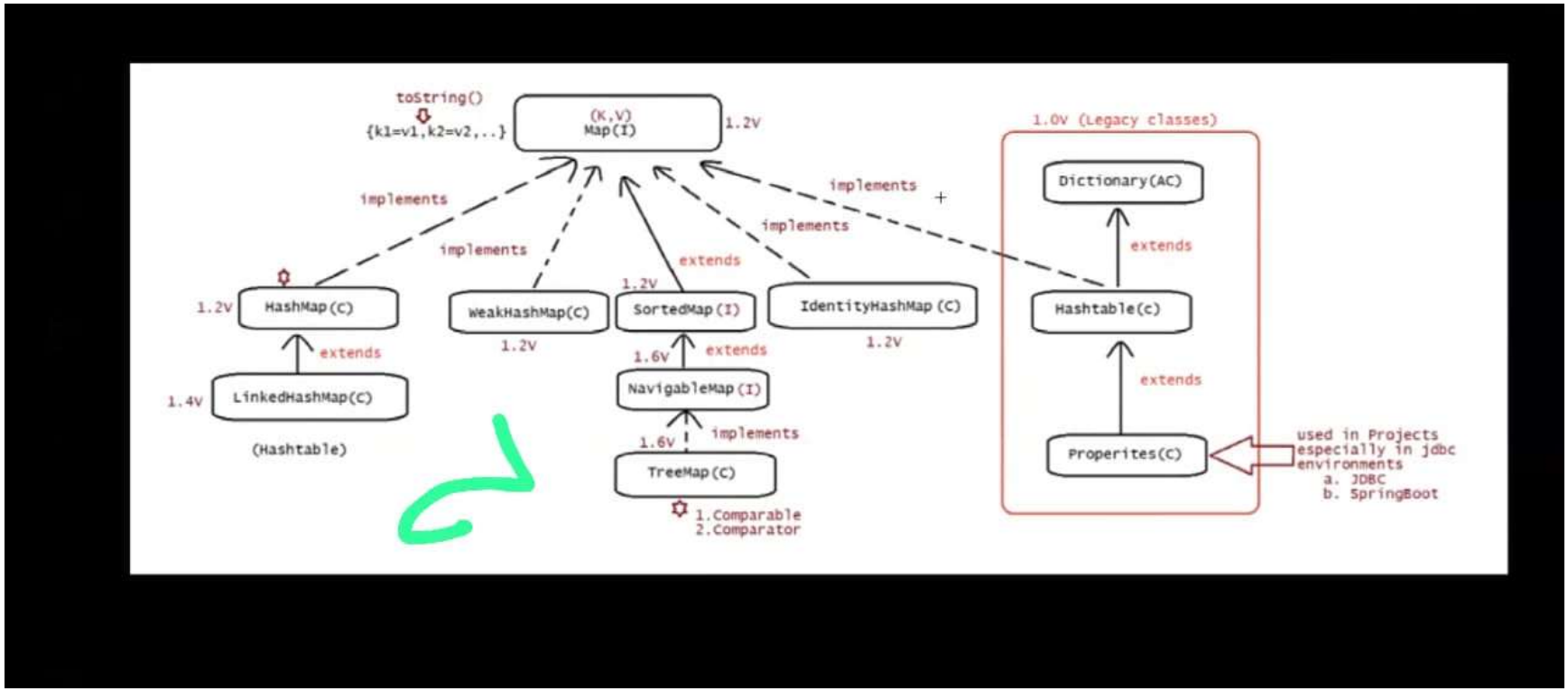
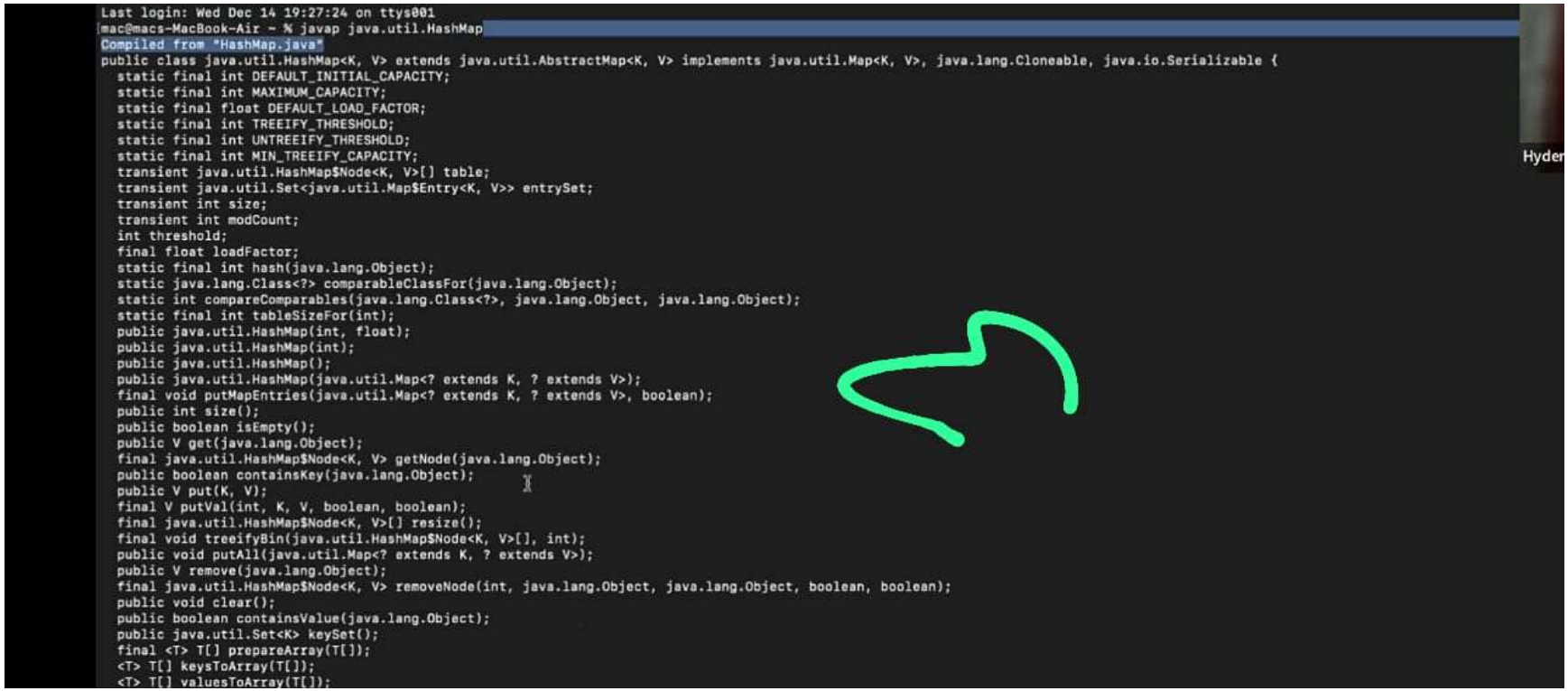
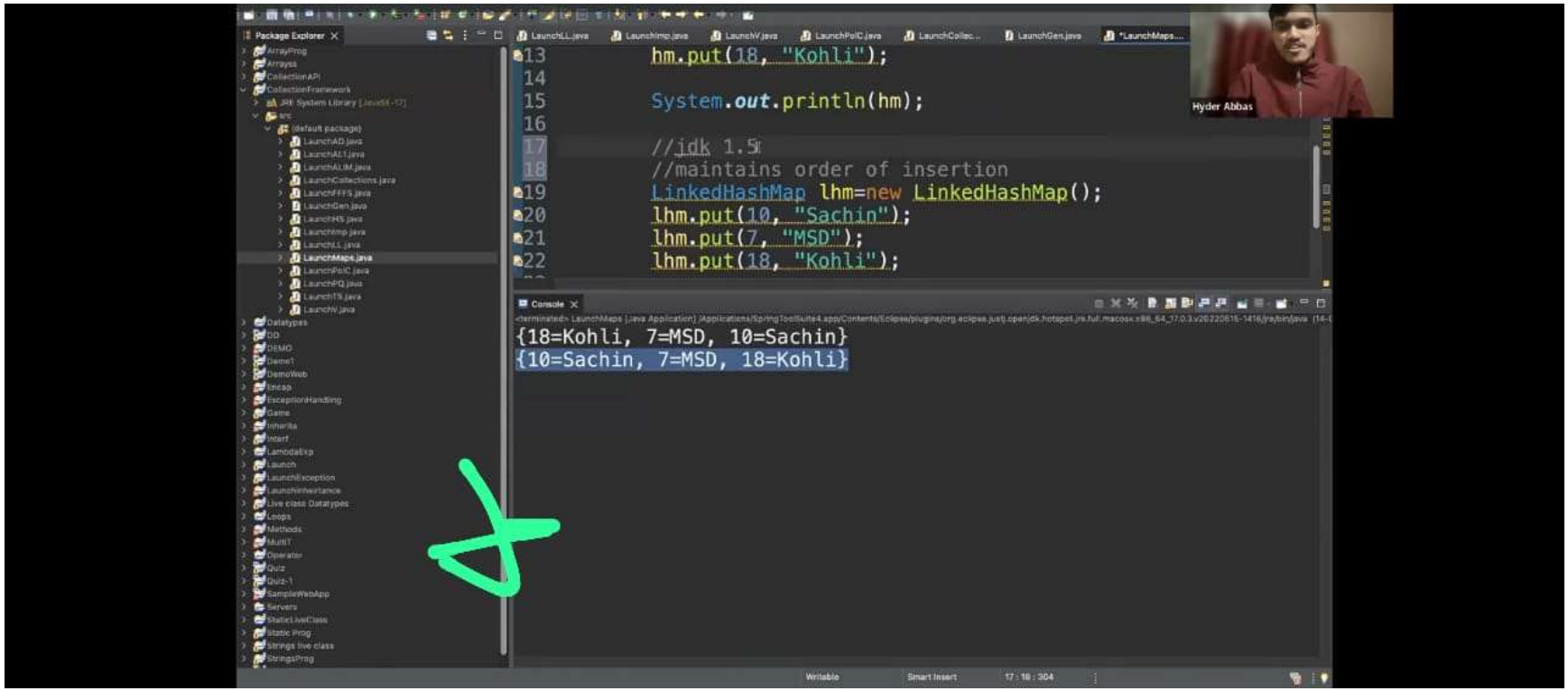
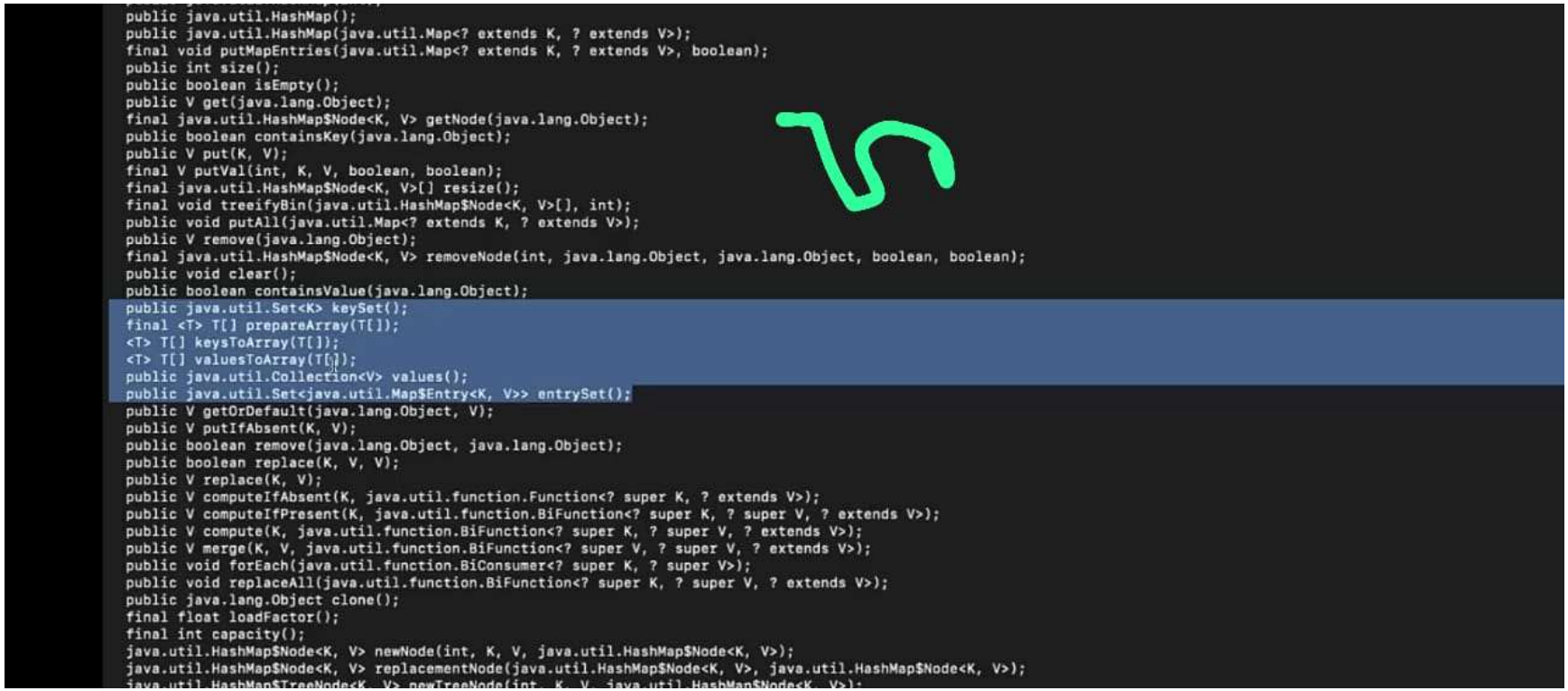
Java Collection Part-4

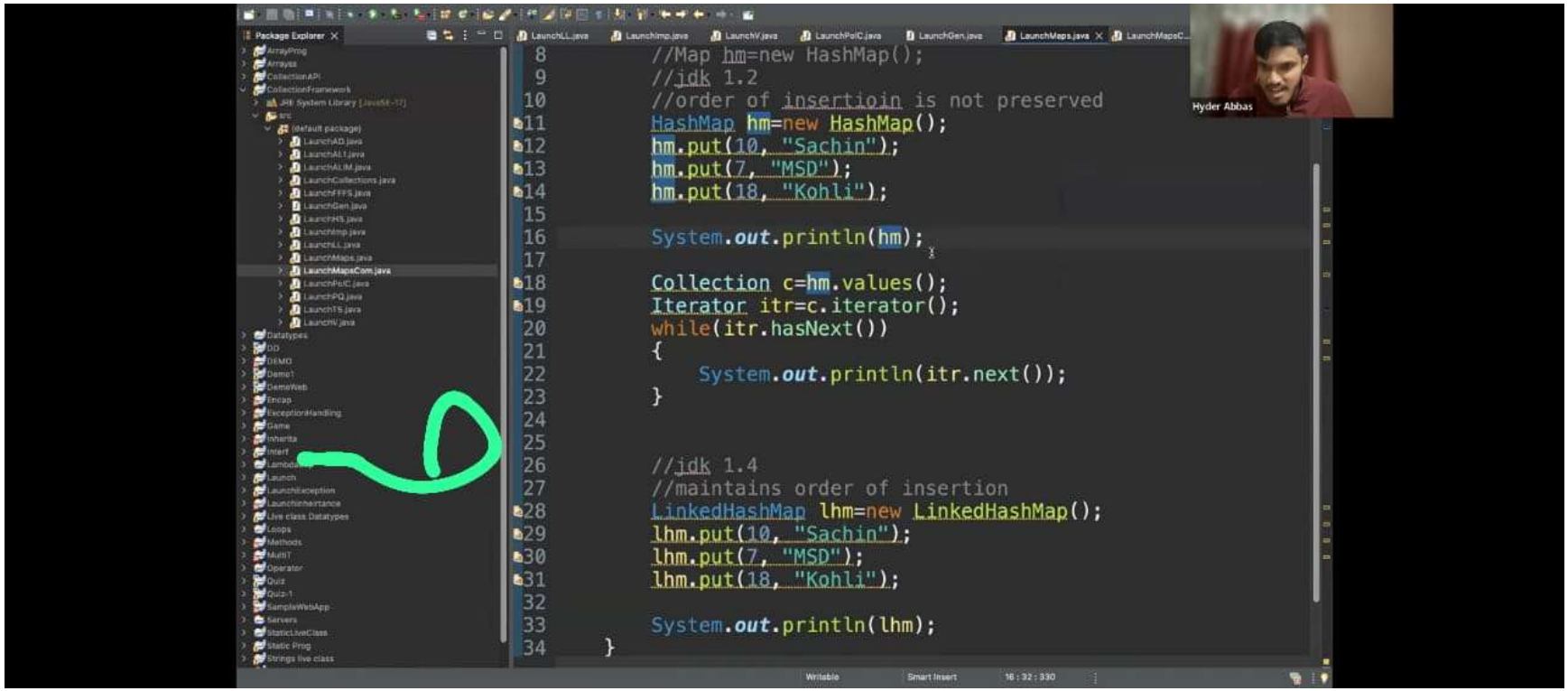


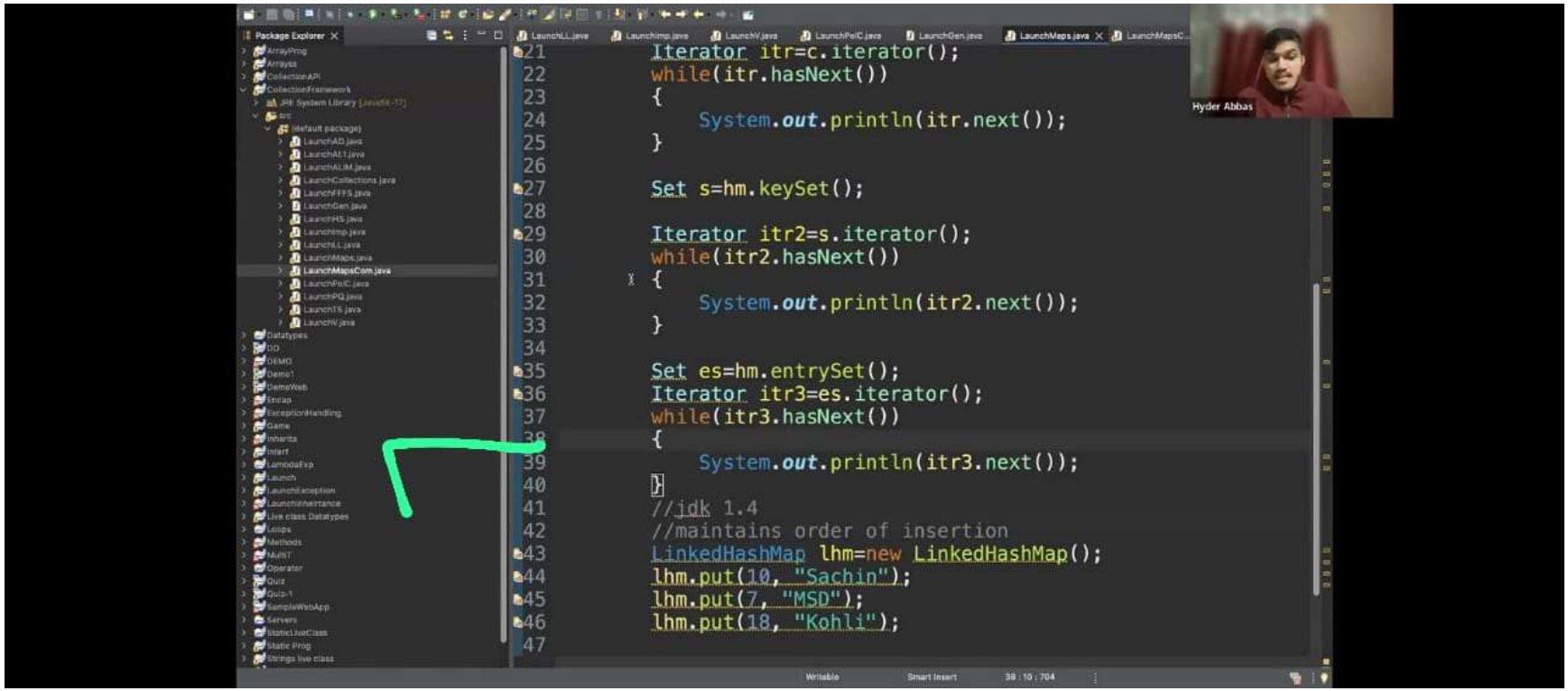

toString()
{k1=v1,k2=v2,..}
(K,V)
Map(I)
1.2v
1.0v (Legacy classes)
implements
implements
implements
implements
extends
1.2v
HashMap(C)
1.2v
WeakHashMap(C)
1.2v
SortedMap(I)
IdentityHashMap(C)
1.2v
Dictionary(AC)
extends
Hashtable(c)
extends
1.4v
LinkedHashMap(C)
(Hashtable)
1.6v
NavigableMap(I)
1.6v
implements
TreeMap(C)
1.Comparable
2.Comparator
Properites(C)
used in Projects
especially in jdbc
environments
a. JDBC
b. SpringBoot

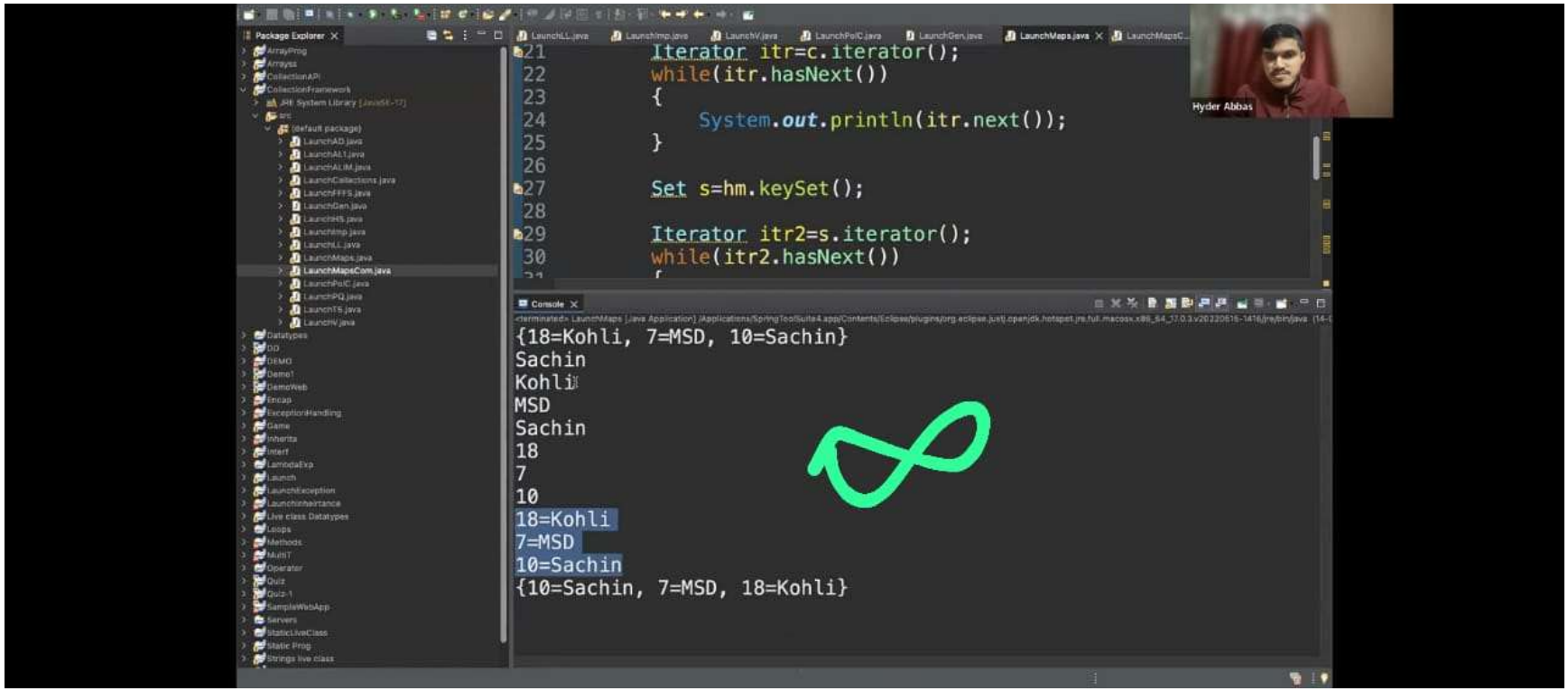

Last login: Wed Dec 14 19:27:24 on ttys001
mac@macs-MacBook-Air ~ % javap java.util.HashMap
Compiled from "HashMap.java"
public class java.util.HashMap<K, V> extends java.util.AbstractMap<K, V> implements java.util.Map<K, V>, java.lang.Cloneable, java.io.Serializable {
static final int DEFAULT_INITIAL_CAPACITY;
static final int MAXIMUM_CAPACITY;
static final float DEFAULT_LOAD_FACTOR;
static final int TREEIFY_THRESHOLD;
static final int UNTREEIFY_THRESHOLD;
static final int MIN_TREEIFY_CAPACITY;
transient java.util.HashMap$Node<K, V>[] table;
transient java.util.Set<java.util.Map$Entry<K, V>> entrySet;
transient int size;
transient int modCount;
int threshold;
final float loadFactor;
static final int hash(java.lang.Object);
static java.lang.Class<?> comparableClassFor(java.lang.Object);
static int compareComparables(java.lang.Class<?>, java.lang.Object, java.lang.Object);
static final int tableSizeFor(int);
public java.util.HashMap(int, float);
public java.util.HashMap(int);
public java.util.HashMap();
public java.util.HashMap(java.util.Map<? extends K, ? extends V>);
final void putMapEntries(java.util.Map<? extends K, ? extends V>, boolean);
public int size();
public boolean isEmpty();
public V get(java.lang.Object);
final java.util.HashMap$Node<K, V> getNode(java.lang.Object);
public boolean containsKey(java.lang.Object);
public V put(K, V);
final V putVal(int, K, V, boolean, boolean);
final java.util.HashMap$Node<K, V>[] resize();
final void treeifyBin(java.util.HashMap$Node<K, V>[], int);
public void putAll(java.util.Map<? extends K, ? extends V>);
public V remove(java.lang.Object);
final java.util.HashMap$Node<K, V> removeNode(int, java.lang.Object, java.lang.Object, boolean, boolean);
public void clear();
public boolean containsValue(java.lang.Object);
public java.util.Set<K> keySet();
final <T> T[] prepareArray(T[]);
<T> T[] keysToArray(T[]);
<T> T[] valuesToArray(T[]);
Hyder

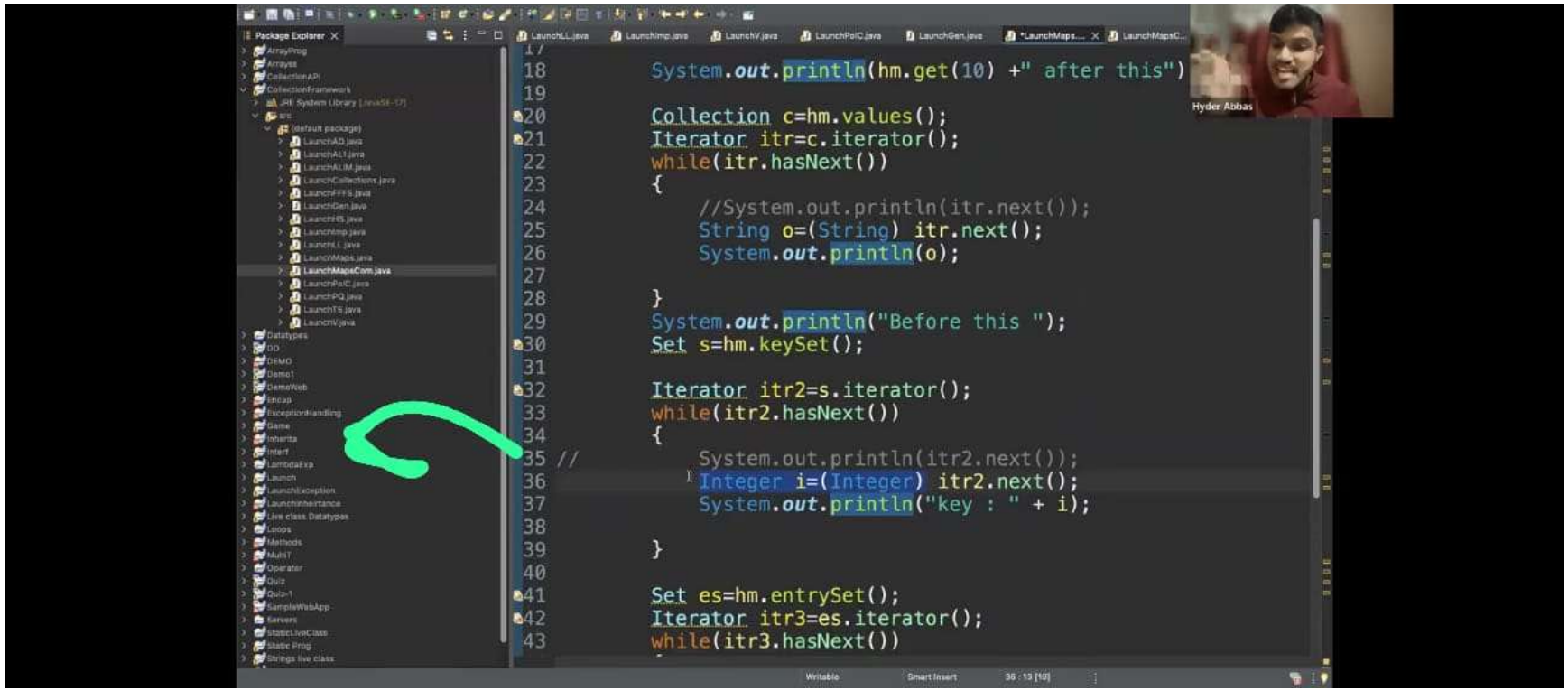

hm.put(18, "Kohli");
System.out.println(hm);
//jdk 1.5
//maintains order of insertion
LinkedHashMap lhm=new LinkedHashMap();
lhm.put(10, "Sachin");
lhm.put(7, "MSD");
lhm.put(18, "Kohli");
{18=Kohli, 7=MSD, 10=Sachin}
{10=Sachin, 7=MSD, 18=Kohli}
Hyder Abbas

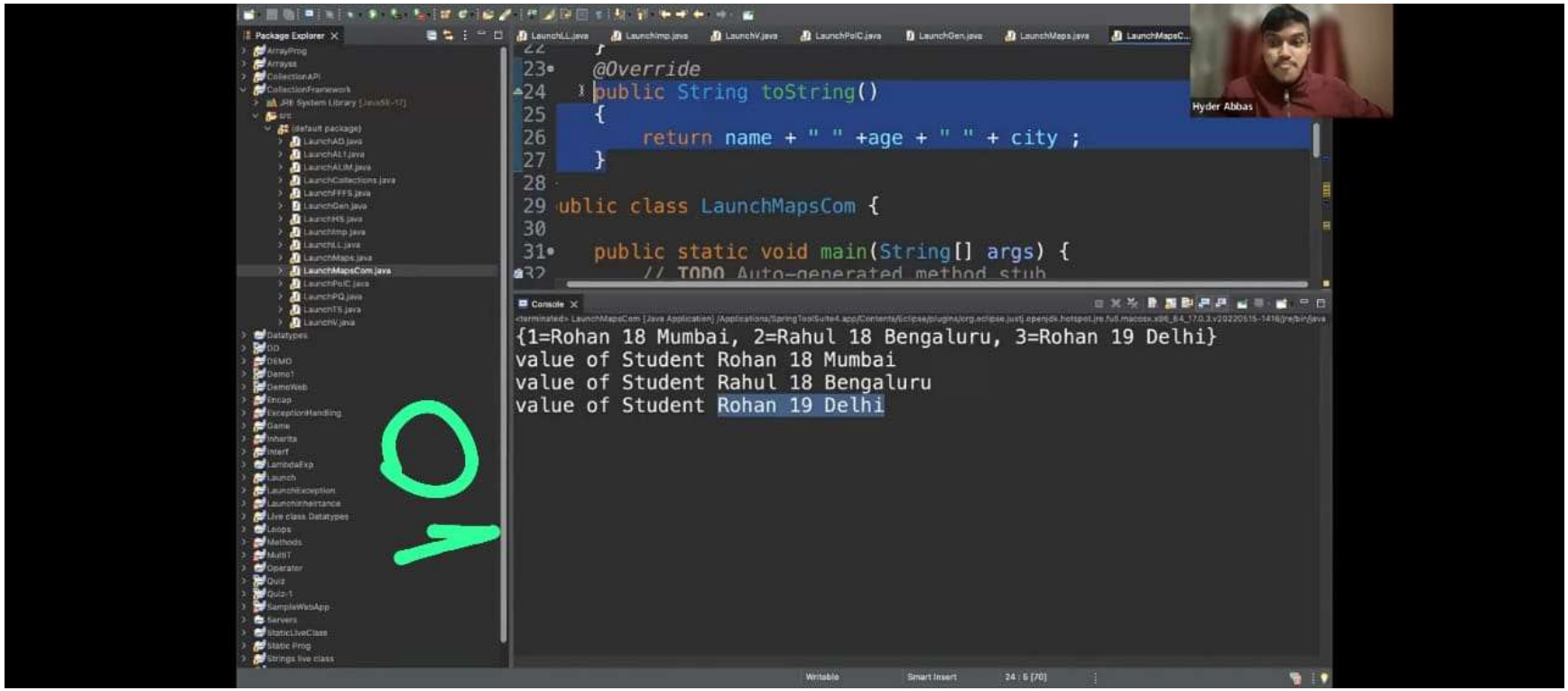

public java.util.HashMap();
public java.util.HashMap(java.util.Map<? extends K, ? extends V>);
final void putMapEntries(java.util.Map<? extends K, ? extends V>, boolean);
public int size();
public boolean isEmpty();
public V get(java.lang.Object);
final java.util.HashMap$Node<K, V> getNode(java.lang.Object);
public boolean containsKey(java.lang.Object);
public V put(K, V);
final V putVal(int, K, V, boolean, boolean);
final java.util.HashMap$Node<K, V>[] resize();
final void treeifyBin(java.util.HashMap$Node<K, V>[], int);
public void putAll(java.util.Map<? extends K, ? extends V>);
public V remove(java.lang.Object);
final java.util.HashMap$Node<K, V> removeNode(int, java.lang.Object, java.lang.Object, boolean, boolean);
public void clear();
public boolean containsValue(java.lang.Object);
public java.util.Set<K> keySet();
final <T> T[] prepareArray(T[]);
<T> T[] keysToArray(T[]);
<T> T[] valuesToArray(T[]);
public java.util.Collection<V> values();
public java.util.Set<java.util.Map$Entry<K, V>> entrySet();
public V getOrDefault(java.lang.Object, V);
public V putIfAbsent(K, V);
public boolean remove(java.lang.Object, java.lang.Object);
public boolean replace(K, V, V);
public V replace(K, V);
public V computeIfAbsent(K, java.util.function.Function<? super K, ? extends V>);
public V computeIfPresent(K, java.util.function.BiFunction<? super K, ? super V, ? extends V>);
public V compute(K, java.util.function.BiFunction<? super K, ? super V, ? extends V>);
public V merge(K, V, java.util.function.BiFunction<? super V, ? super V, ? extends V>);
public void forEach(java.util.function.BiConsumer<? super K, ? super V>);
public void replaceAll(java.util.function.BiFunction<? super K, ? super V, ? extends V>);
public java.lang.Object clone();
final float loadFactor();
final int capacity();
java.util.HashMap$Node<K, V> newNode(int, K, V, java.util.HashMap$Node<K, V>);
java.util.HashMap$Node<K, V> replacementNode(java.util.HashMap$Node<K, V>, java.util.HashMap$Node<K, V>);

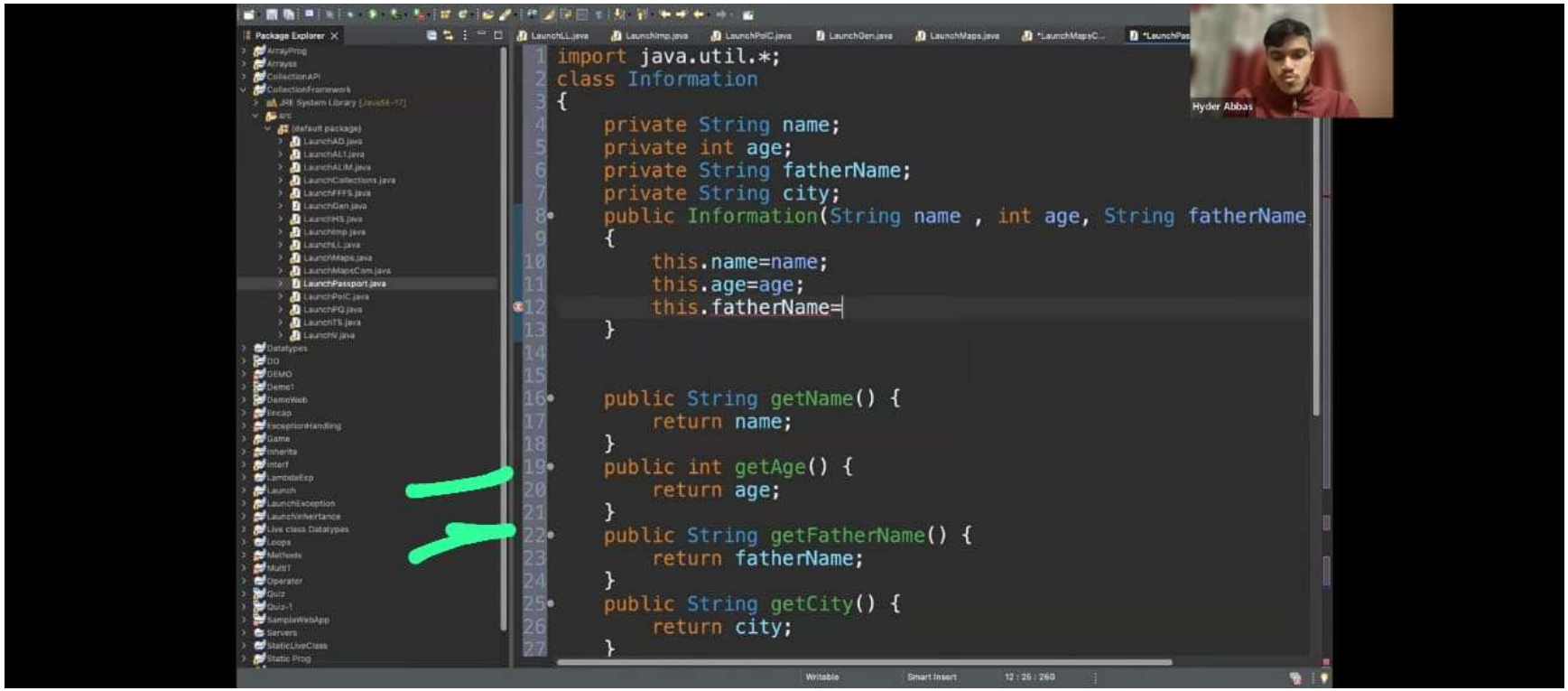

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchV.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
LaunchMapsC...
Hyder Abbas
//Map hm=new HashMap();
//jdk 1.2
//order of insertioin is not preserved
HashMap hm=new HashMap();
hm.put(10, "Sachin");
hm.put(7, "MSD");
hm.put(18, "Kohli");
System.out.println(hm);
Collection c=hm.values();
Iterator itr=c.iterator();
while(itr.hasNext())
{
System.out.println(itr.next());
}
//jdk 1.4
//maintains order of insertion
LinkedHashMap lhm=new LinkedHashMap();
lhm.put(10, "Sachin");
lhm.put(7, "MSD");
lhm.put(18, "Kohli");
System.out.println(lhm);
}
Writable
Smart Insert
16 : 32 : 330

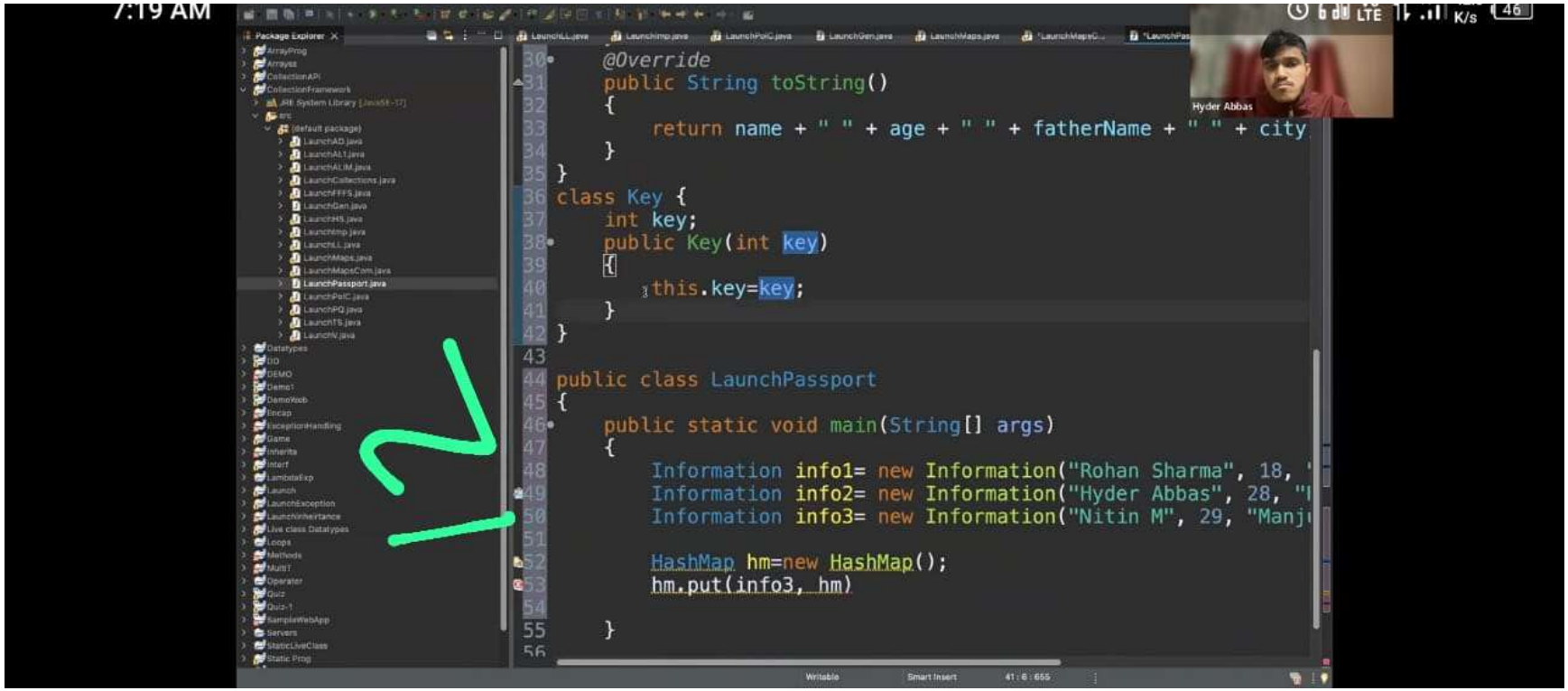

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchV.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
LaunchMapsC...
Hyder Abbas
Iterator itr=c.iterator();
while(itr.hasNext())
{
System.out.println(itr.next());
}
Set s=hm.keySet();
Iterator itr2=s.iterator();
while(itr2.hasNext())
{
System.out.println(itr2.next());
}
Set es=hm.entrySet();
Iterator itr3=es.iterator();
while(itr3.hasNext())
{
System.out.println(itr3.next());
}
//jdk 1.4
//maintains order of insertion
LinkedHashMap lhm=new LinkedHashMap();
lhm.put(10, "Sachin");
lhm.put(7, "MSD");
lhm.put(18, "Kohli");
Writable
Smart Insert
38 : 10 : 704

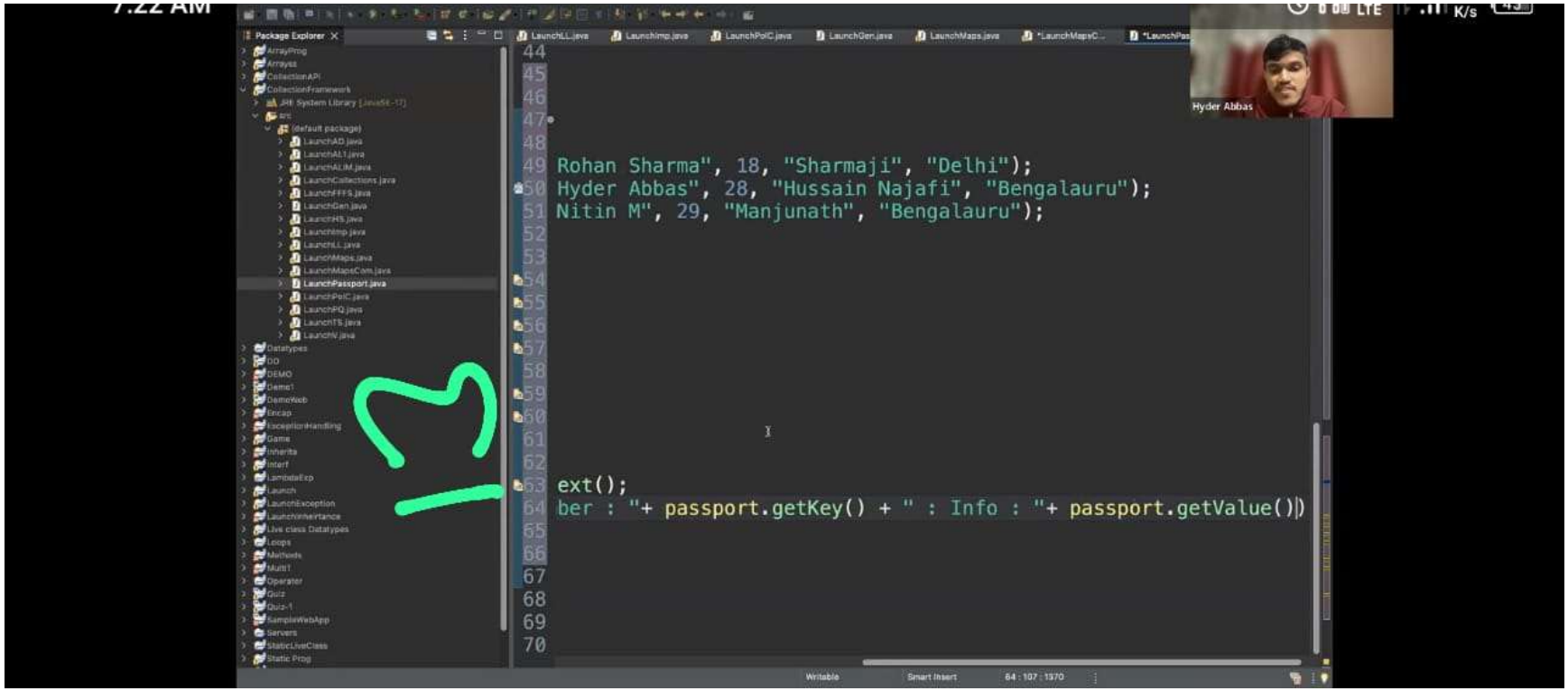

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchV.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
Iterator itr=c.iterator();
while(itr.hasNext())
{
System.out.println(itr.next());
}
Set s=hm.keySet();
Iterator itr2=s.iterator();
while(itr2.hasNext())
Hyder Abbas
Console
{18=Kohli, 7=MSD, 10=Sachin}
Sachin
Kohli
MSD
Sachin
18
7
10
18=Kohli
7=MSD
10=Sachin
{10=Sachin, 7=MSD, 18=Kohli}

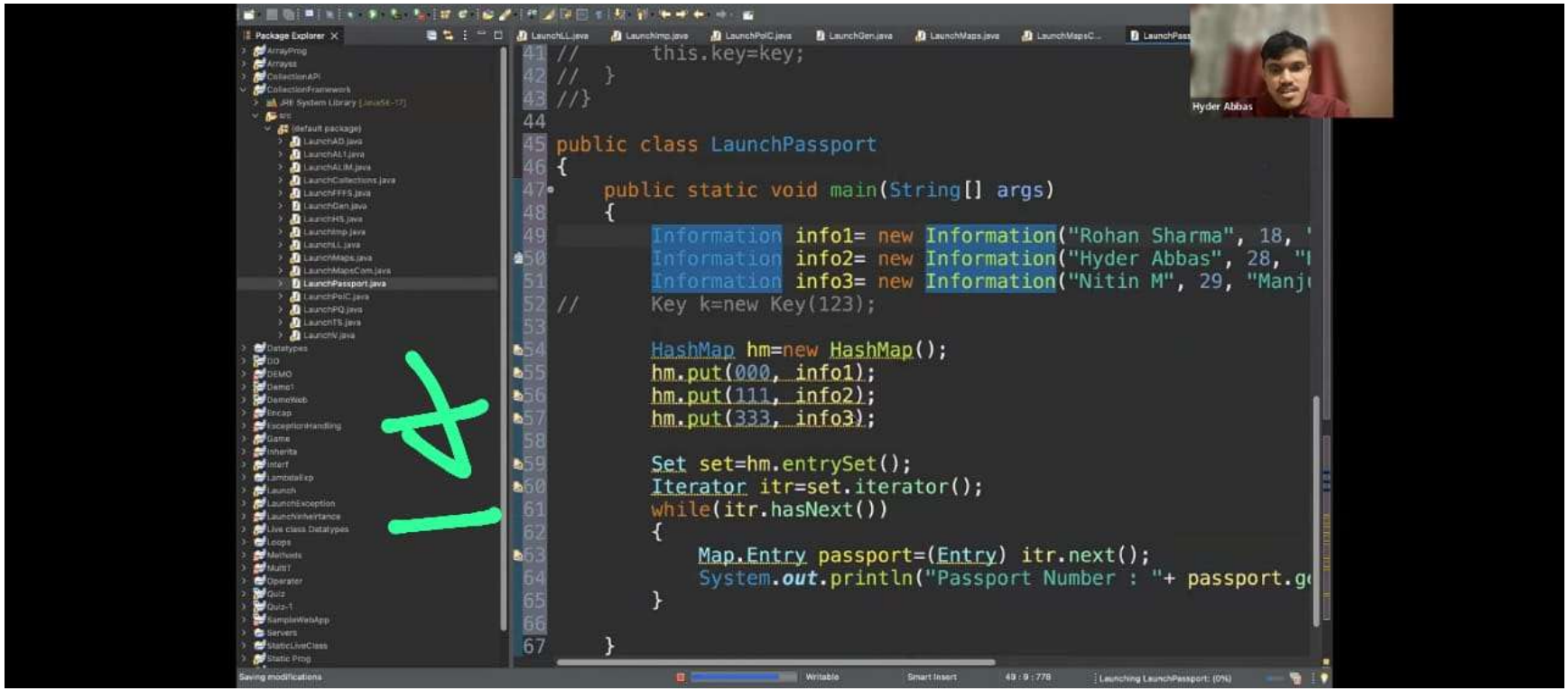

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchV.java
LaunchPolC.java
LaunchGen.java
*LaunchMaps...
LaunchMapsC...
Hyder Abbas
System.out.println(hm.get(10) +" after this")
Collection c=hm.values();
Iterator itr=c.iterator();
while(itr.hasNext())
{
//System.out.println(itr.next());
String o=(String) itr.next();
System.out.println(o);
}
System.out.println("Before this ");
Set s=hm.keySet();
Iterator itr2=s.iterator();
while(itr2.hasNext())
{
// System.out.println(itr2.next());
Integer i=(Integer) itr2.next();
System.out.println("key : " + i);
}
Set es=hm.entrySet();
Iterator itr3=es.iterator();
while(itr3.hasNext())
Writable
Smart Insert
36 : 13 [19]

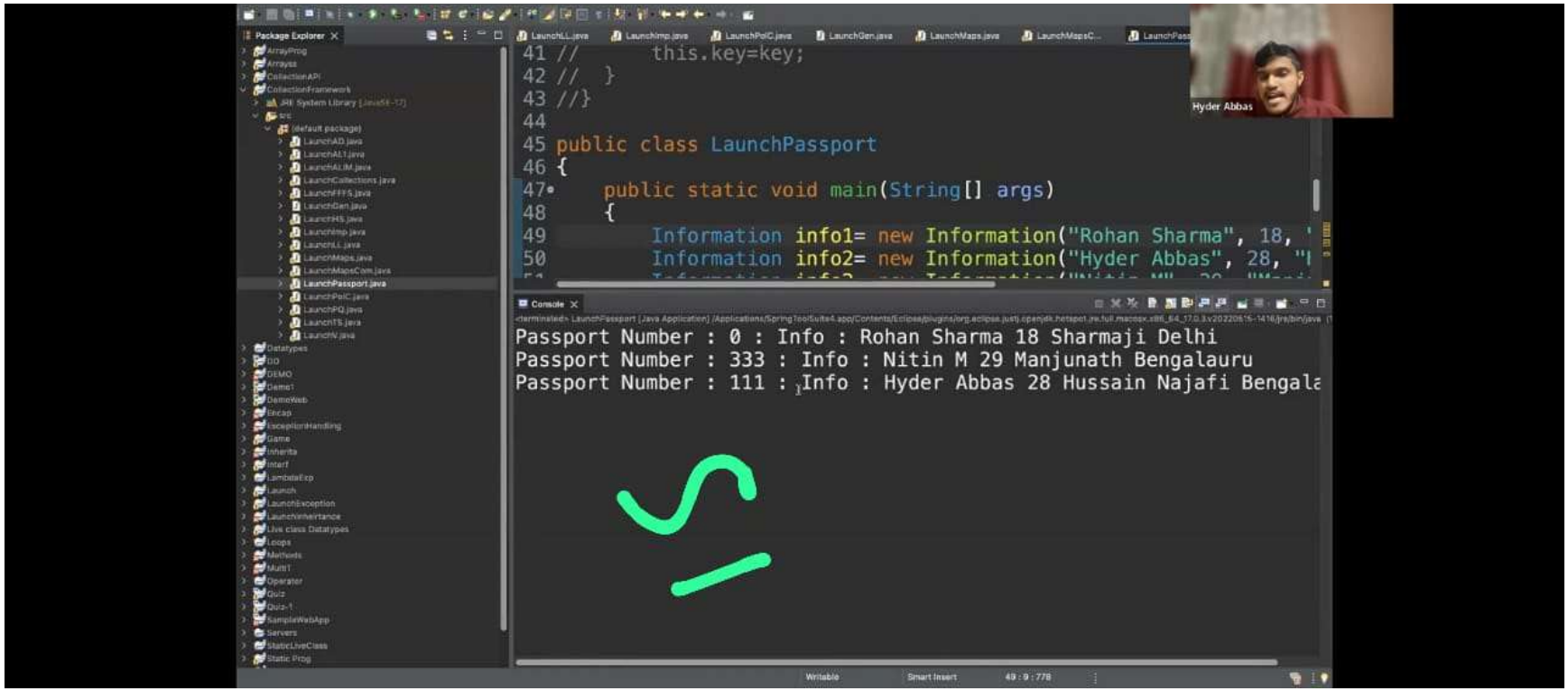

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchV.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
LaunchMapsC...
@Override
public String toString()
{
return name + " " +age + " " + city ;
}
ublic class LaunchMapsCom {
public static void main(String[] args) {
// TODO Auto-generated method stub
Console
{1=Rohan 18 Mumbai, 2=Rahul 18 Bengaluru, 3=Rohan 19 Delhi}
value of Student Rohan 18 Mumbai
value of Student Rahul 18 Bengaluru
value of Student Rohan 19 Delhi
Hyder Abbas
Writable
Smart Insert
24 : 6 [70]

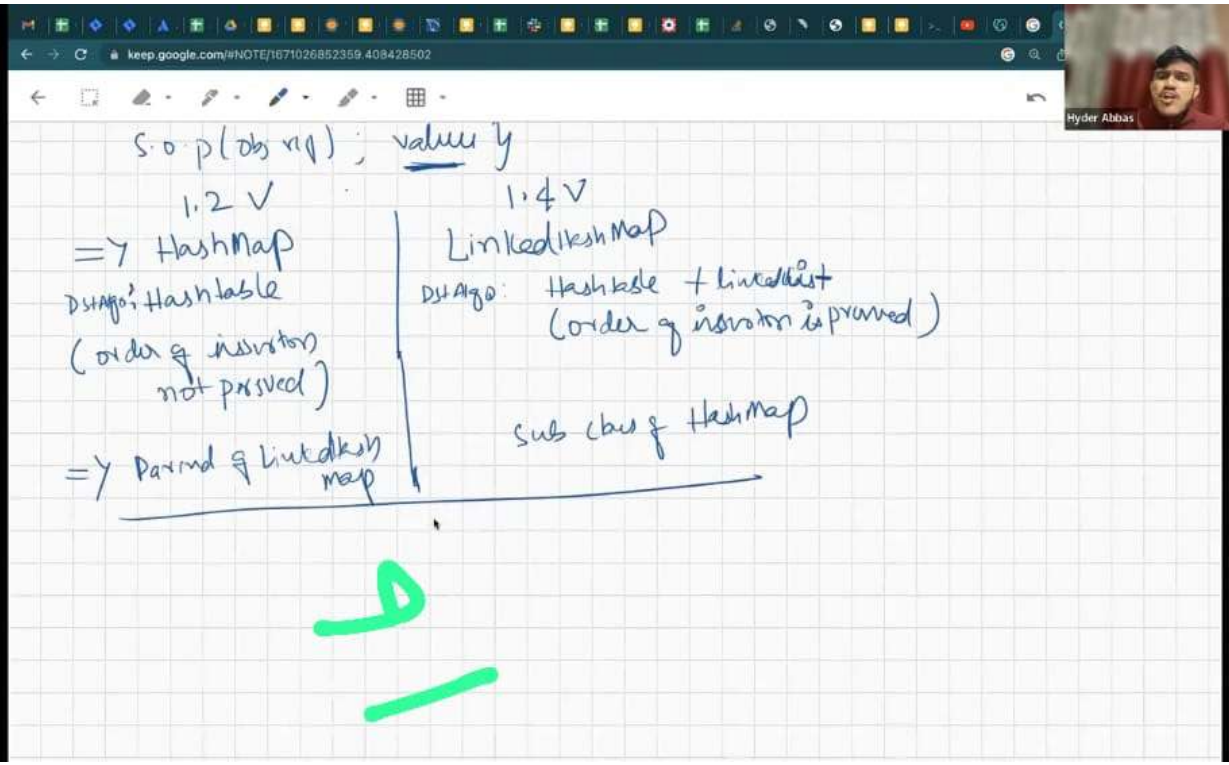

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
Hyder Abbas
import java.util.*;
class Information
{
private String name;
private int age;
private String fatherName;
private String city;
public Information(String name , int age, String fatherName
{
this.name=name;
this.age=age;
this.fatherName=
}
public String getName() {
return name;
}
public int getAge() {
return age;
}
public String getFatherName() {
return fatherName;
}
public String getCity() {
return city;
}
Writable
Smart Insert
12 : 35 : 260

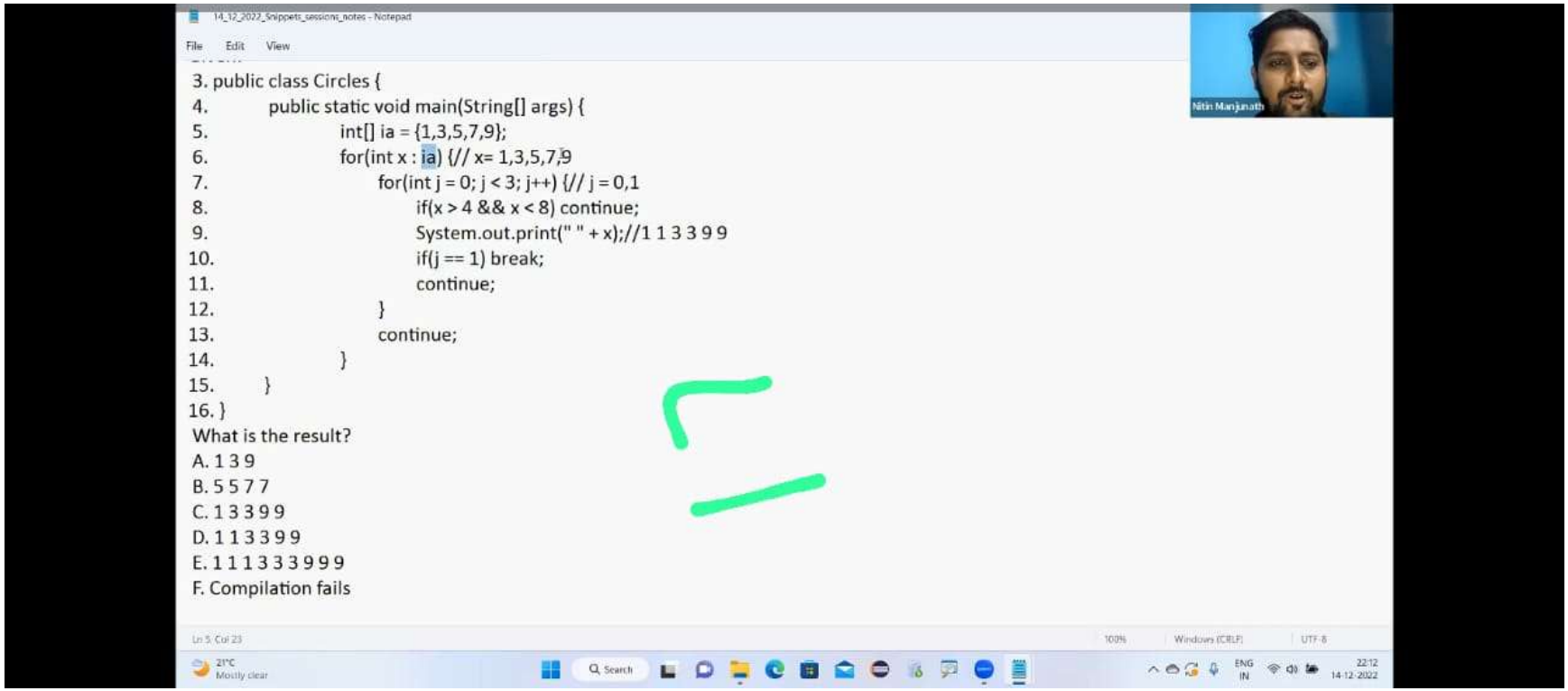

7:19 AM
Hyder Abbas
@Override
public String toString()
{
return name + " " + age + " " + fatherName + " " + city
}
}
class Key {
int key;
public Key(int key)
{
this.key=key;
}
}
public class LaunchPassport
{
public static void main(String[] args)
{
Information info1= new Information("Rohan Sharma", 18,
Information info2= new Information("Hyder Abbas", 28,
Information info3= new Information("Nitin M", 29, "Manj
HashMap hm=new HashMap();
hm.put(info3, hm)
}

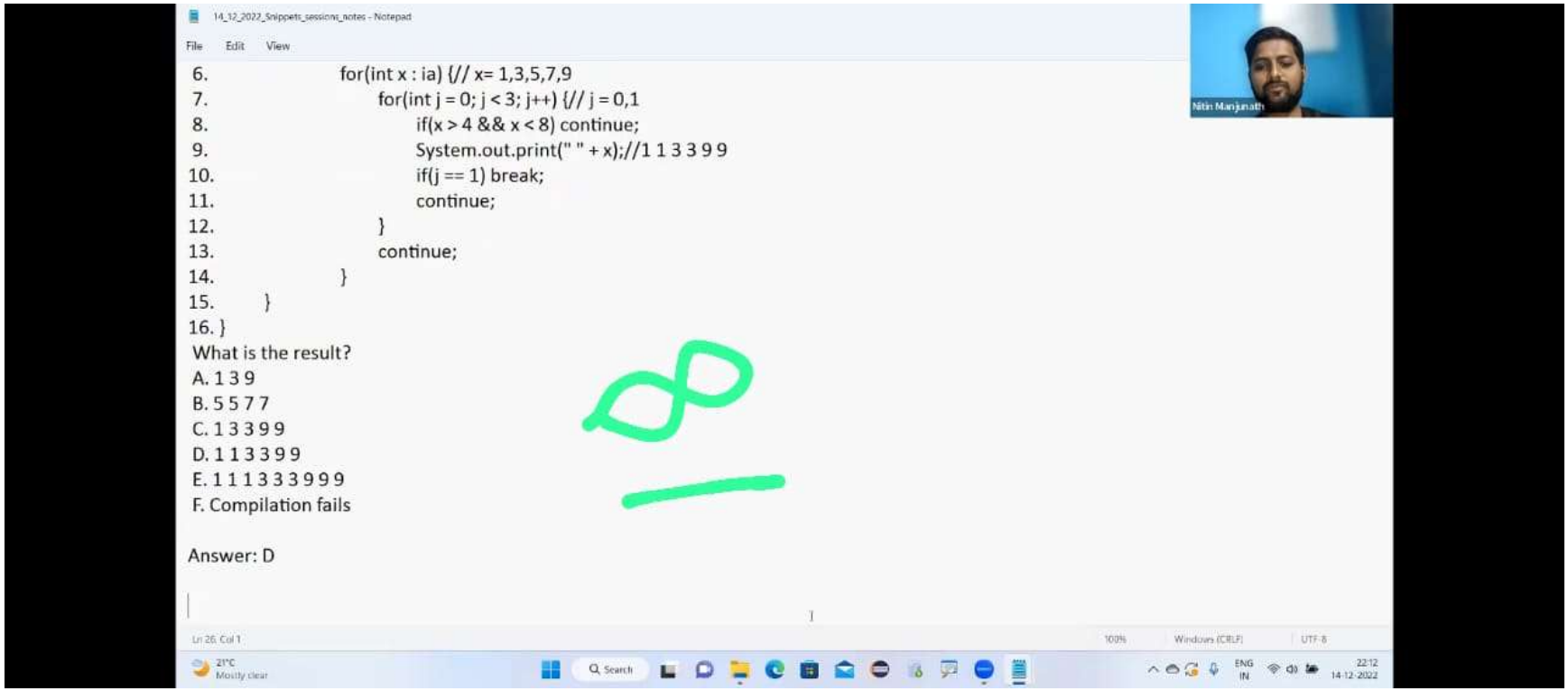

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
Hyder Abbas
Rohan Sharma", 18, "Sharmaji", "Delhi");
Hyder Abbas", 28, "Hussain Najafi", "Bengalauru");
Nitin M", 29, "Manjunath", "Bengalauru");
ext();
ber : "+ passport.getKey() + " : Info : "+ passport.getValue())
Writable
Smart Insert
64 : 107 : 1370

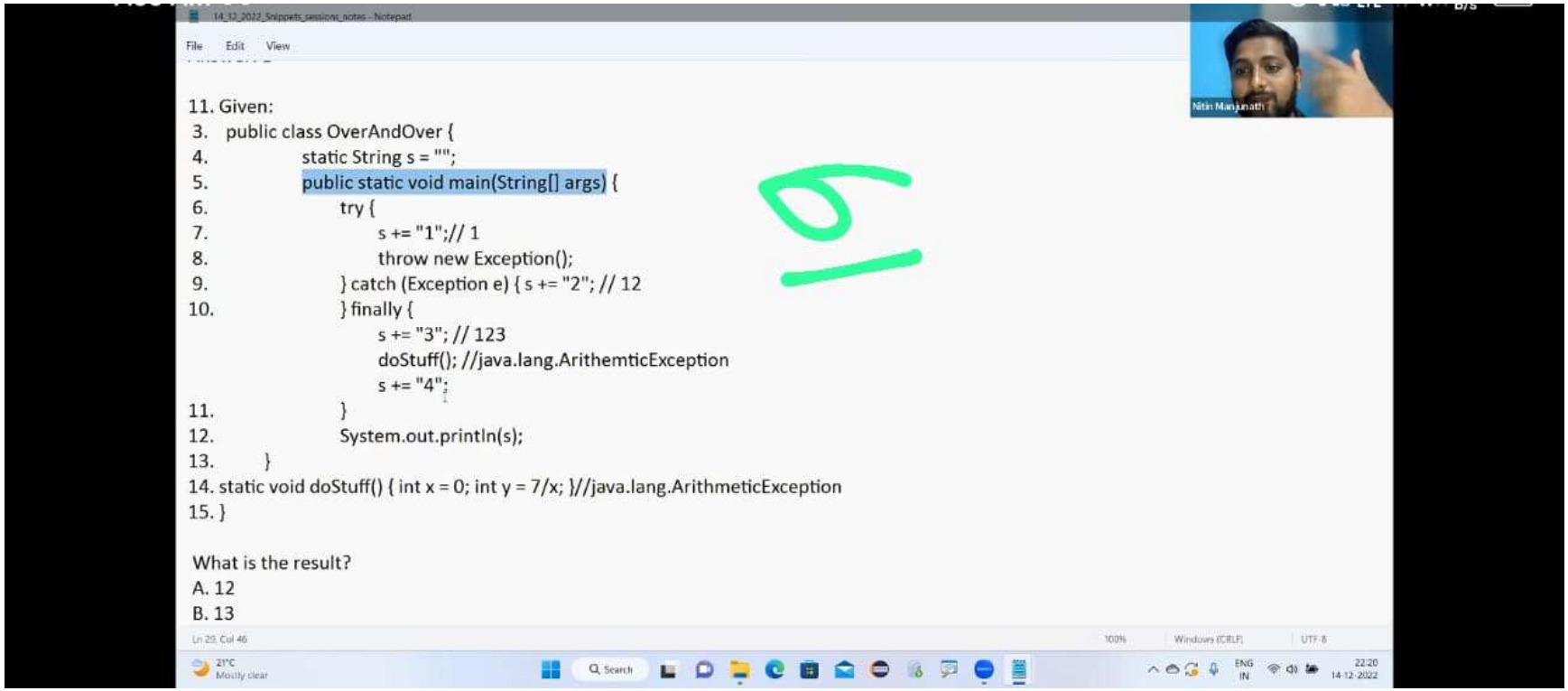

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
LaunchPassport.java
Hyder Abbas
14
// this.key=key;
// }
//}
public class LaunchPassport
{
public static void main(String[] args)
{
Information info1= new Information("Rohan Sharma", 18,
Information info2= new Information("Hyder Abbas", 28,
Information info3= new Information("Nitin M", 29, "Manj
// Key k=new Key(123);
HashMap hm=new HashMap();
hm.put(000, info1);
hm.put(111, info2);
hm.put(333, info3);
Set set=hm.entrySet();
Iterator itr=set.iterator();
while(itr.hasNext())
{
Map.Entry passport=(Entry) itr.next();
System.out.println("Passport Number : "+ passport.g
}
}
Saving modifications
Writable
Smart Insert
49 : 9 : 778
Launching LaunchPassport: (0%)

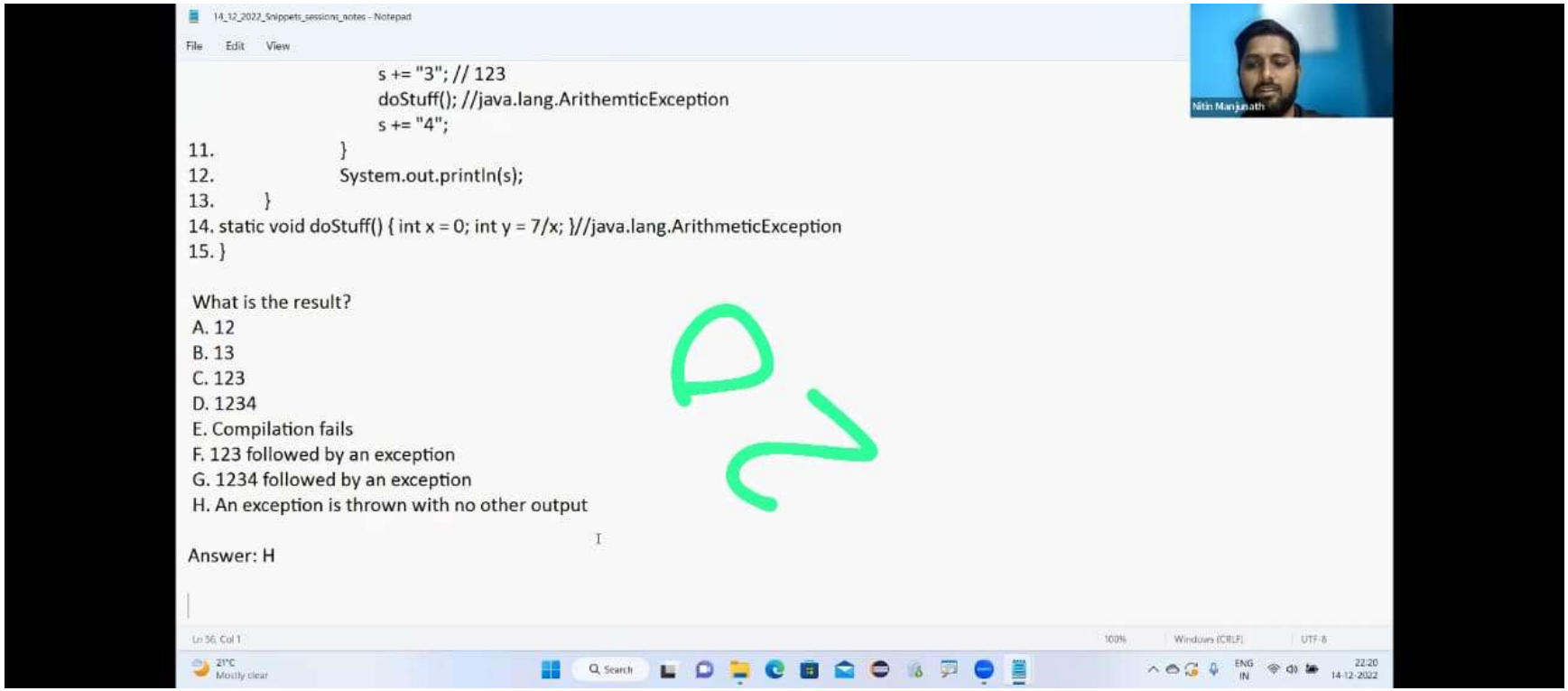

Package Explorer
LaunchLL.java
LaunchImp.java
LaunchPolC.java
LaunchGen.java
LaunchMaps.java
LaunchPassport.java
41 // this.key=key;
42 // }
43 //}
44
45 public class LaunchPassport
46 {
47• public static void main(String[] args)
48 {
49 Information info1= new Information("Rohan Sharma", 18, '
50 Information info2= new Information("Hyder Abbas", 28, "I
Hyder Abbas
Console
Passport Number : 0 : Info : Rohan Sharma 18 Sharmaji Delhi
Passport Number : 333 : Info : Nitin M 29 Manjunath Bengalauru
Passport Number : 111 : Info : Hyder Abbas 28 Hussain Najafi Bengala
Writable
Smart Insert
49 : 9 : 778

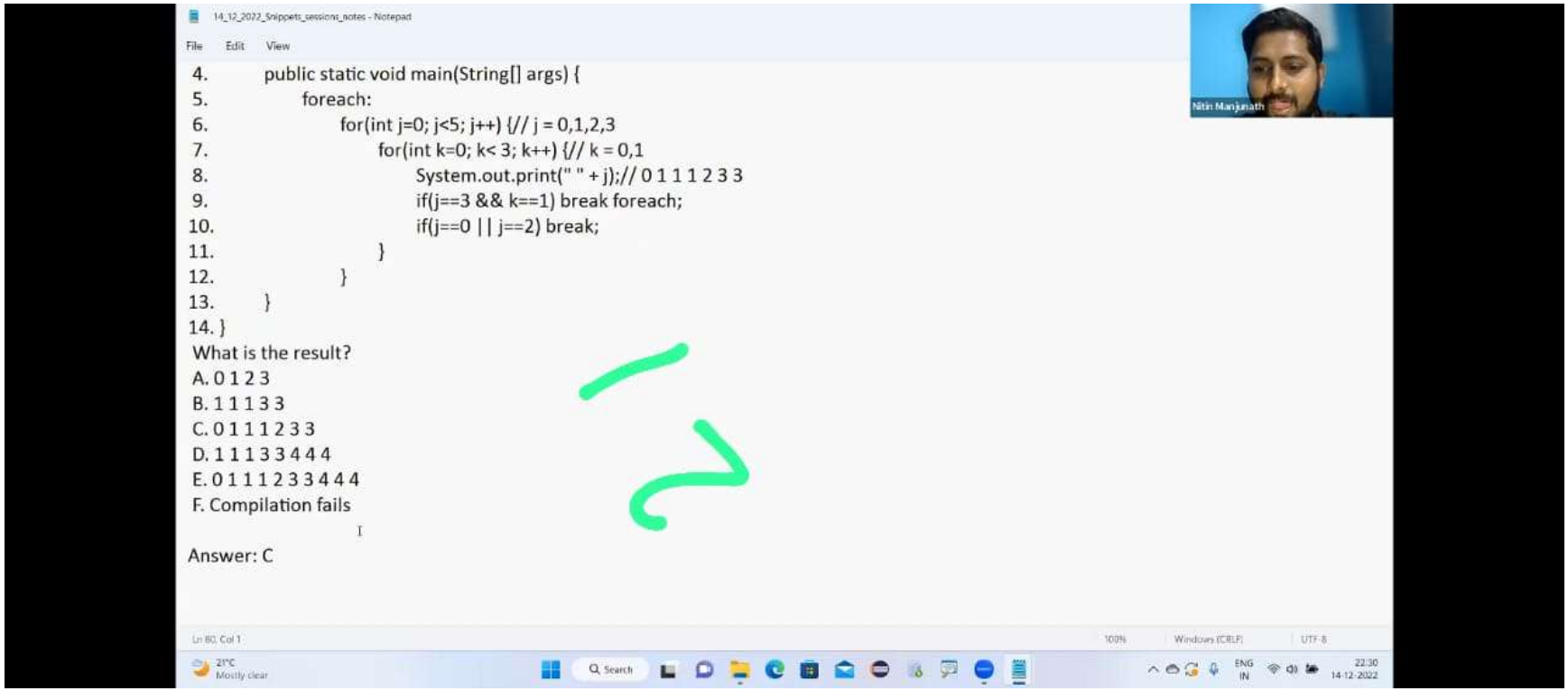

keep.google.com/#NOTE/1671026852359.408428502
Hyder Abbas
S.o p(obj ref) ; value y
1.2 V
=> HashMap
DS Algo: Hashtable
(order & insertion not preserved)
=> Parent of LinkedHashMap
1.4 V
LinkedHashMap
DS Algo: Hashtable + LinkedList
(order of insertion is preserved)
Sub class of HashMap

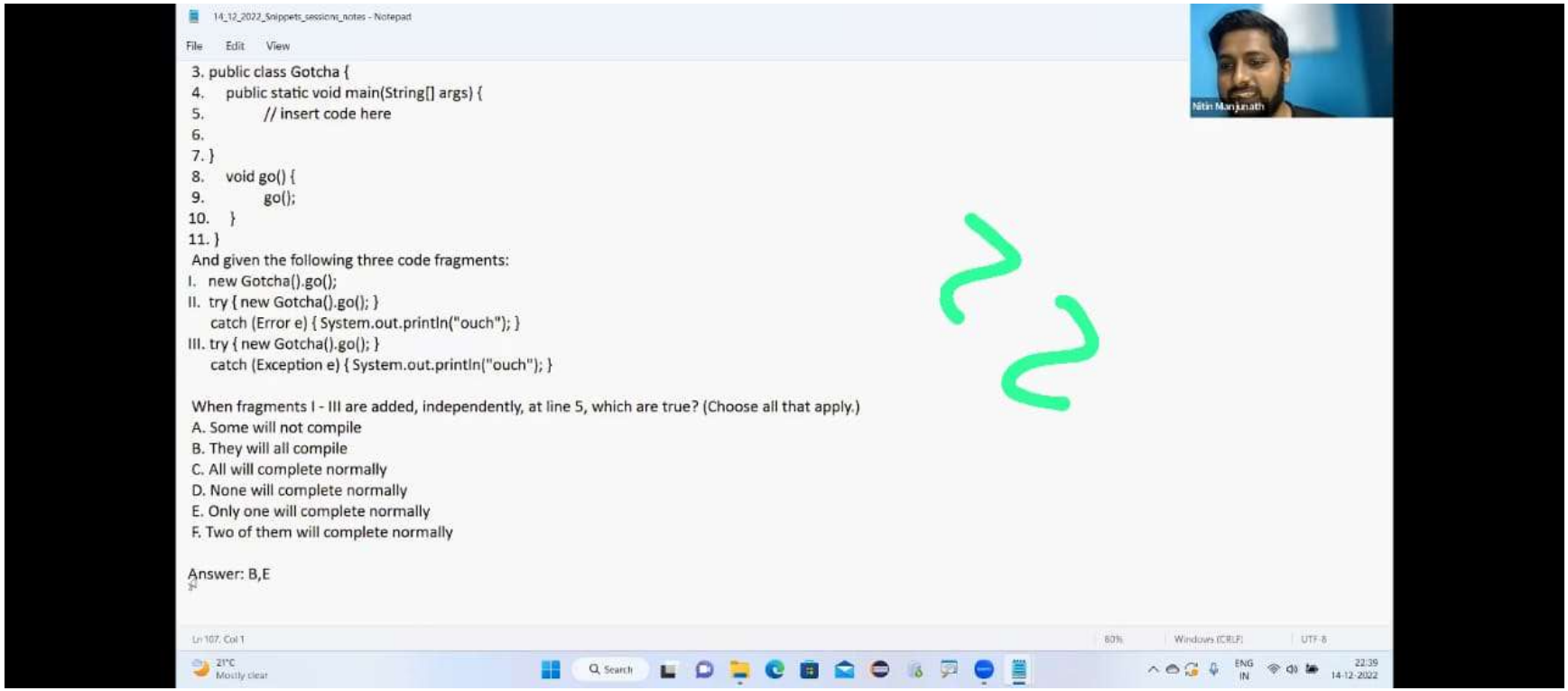

14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
3. public class Circles {
4. public static void main(String[] args) {
5. int[] ia = {1,3,5,7,9};
6. for(int x : ia) {// x= 1,3,5,7,9
7. for(int j = 0; j < 3; j++) {// j = 0,1
8. if(x > 4 && x < 8) continue;
9. System.out.print(" " + x);//1 1 3 3 9 9
10. if(j == 1) break;
11. continue;
12. }
13. continue;
14. }
15. }
16. }
What is the result?
A. 1 3 9
B. 5 5 7 7
C. 1 3 3 9 9
D. 1 1 3 3 9 9
E. 1 1 1 3 3 3 9 9 9
F. Compilation fails
Nitin Manjunath
Ln 5, Col 23
100%
Windows (CRLF)
UTF-8
21°C
Mostly clear
Search
ENG
IN
22:12
14-12-2022

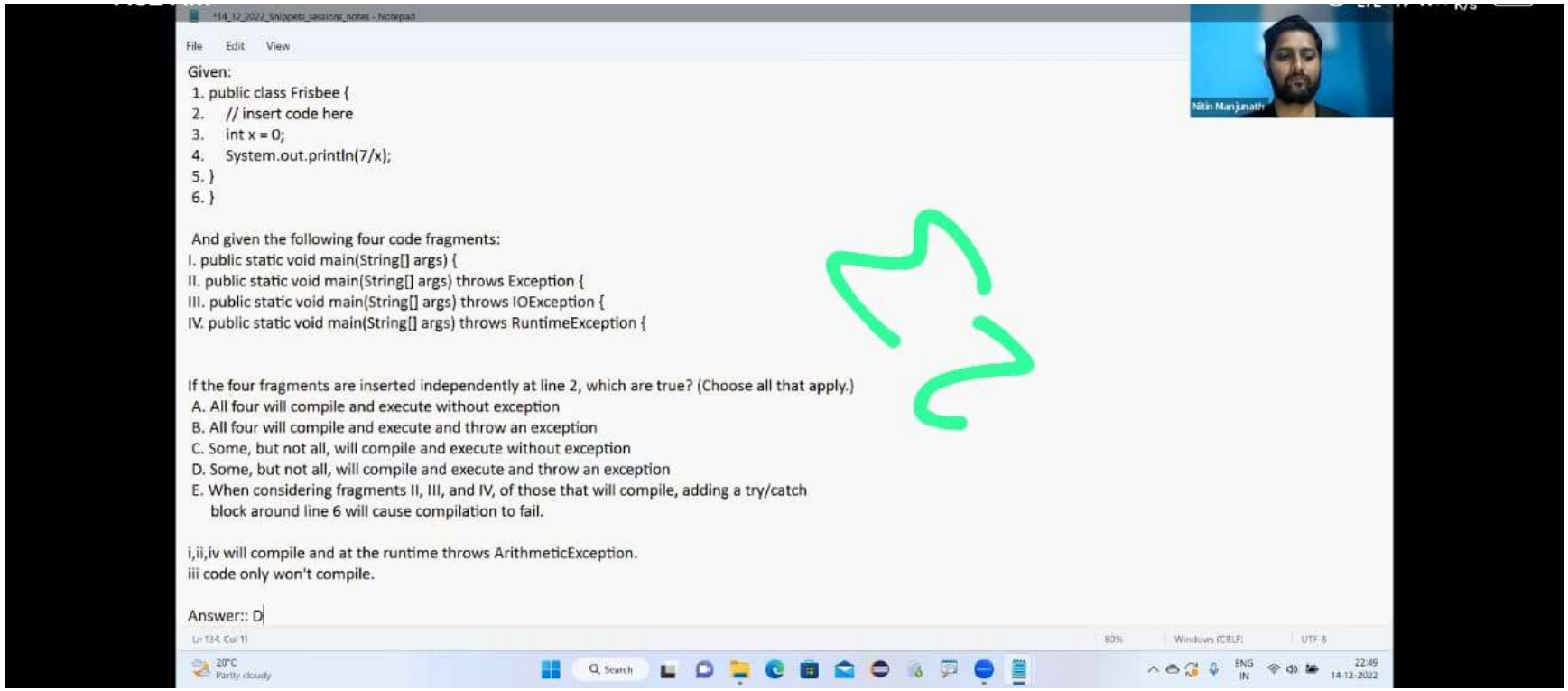

14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
6. for(int x : ia) {// x= 1,3,5,7,9
7. for(int j = 0; j < 3; j++) {// j = 0,1
8. if(x > 4 && x < 8) continue;
9. System.out.print(" " + x);//1 1 3 3 9 9
10. if(j == 1) break;
11. continue;
12. }
13. continue;
14. }
15. }
16. }
What is the result?
A. 1 3 9
B. 5 5 7 7
C. 1 3 3 9 9
D. 1 1 3 3 9 9
E. 1 1 1 3 3 3 9 9 9
F. Compilation fails
Answer: D
Nitin Manjunath
Ln 26, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Mostly clear
Search
ENG
IN
22:12
14-12-2022

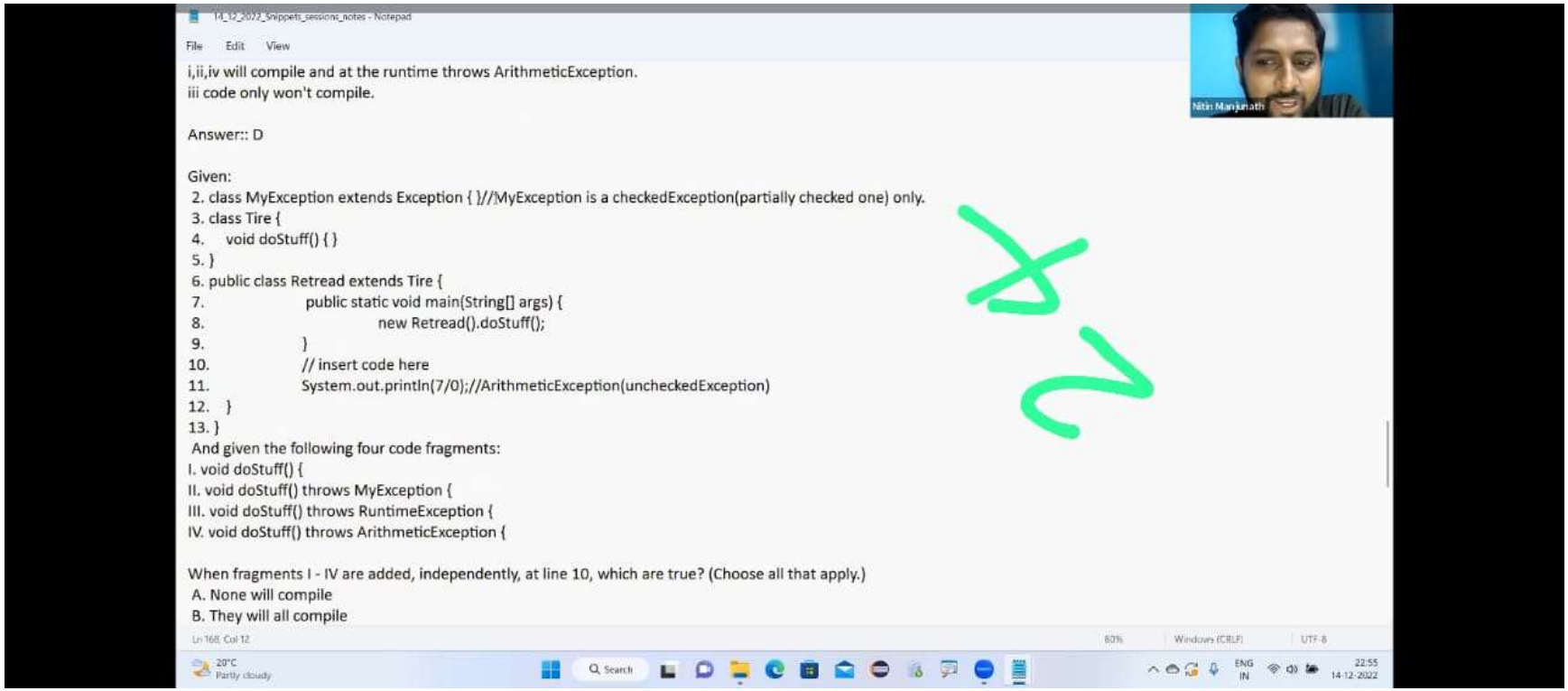

14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
11. Given:
3. public class OverAndOver {
4. static String s = "";
5. public static void main(String[] args) {
6. try {
7. s += "1";// 1
8. throw new Exception();
9. } catch (Exception e) { s += "2"; // 12
10. } finally {
s += "3"; // 123
doStuff(); //java.lang.ArithemticException
s += "4";
11. }
12. System.out.println(s);
13. }
14. static void doStuff() { int x = 0; int y = 7/x; }//java.lang.ArithmeticException
15. }
What is the result?
A. 12
B. 13
Nitin Manjunath
Ln 29, Col 46
100%
Windows (CRLF)
UTF-8
21°C
Mostly clear
Search
ENG
IN
22:20
14-12-2022

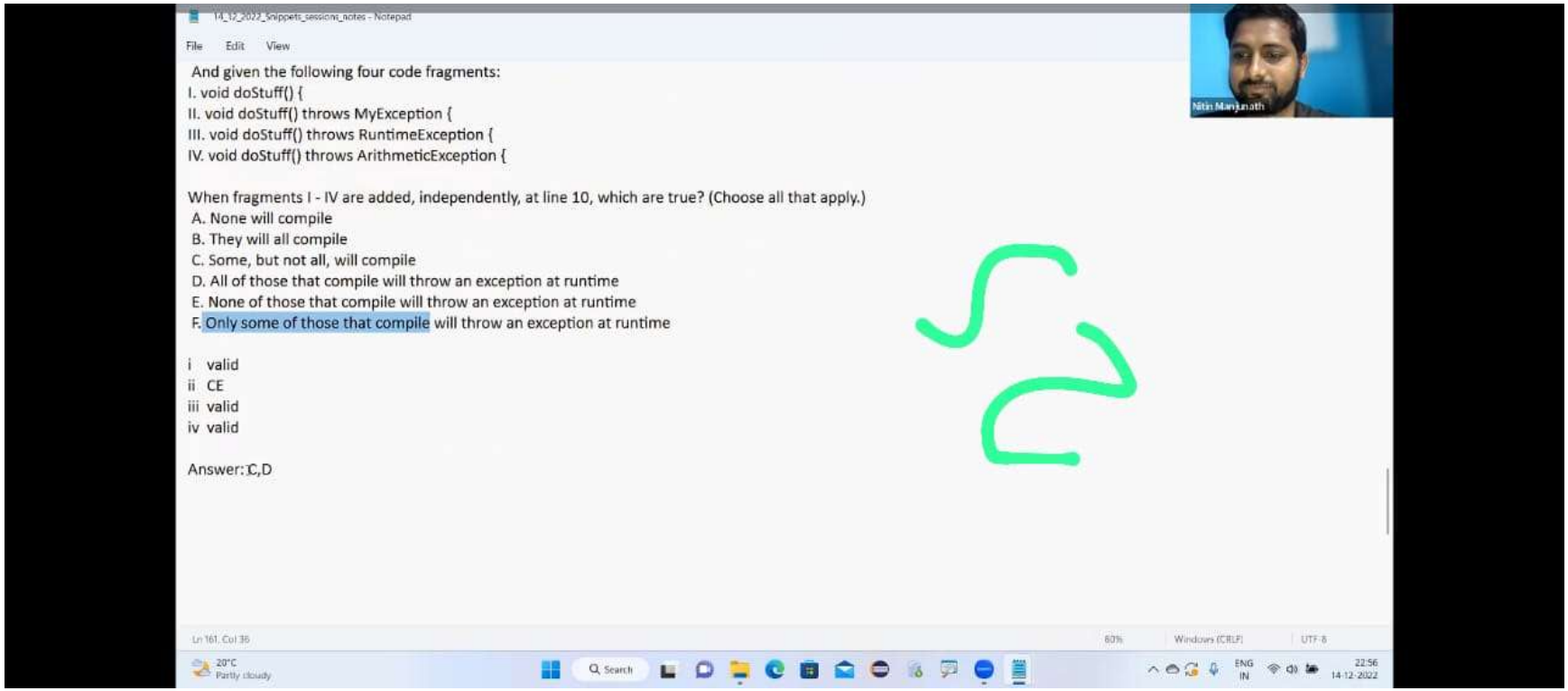

14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
s += "3"; // 123
doStuff(); //java.lang.ArithemticException
s += "4";
}
System.out.println(s);
}
14. static void doStuff() { int x = 0; int y = 7/x; }//java.lang.ArithmeticException
15. }
What is the result?
A. 12
B. 13
C. 123
D. 1234
E. Compilation fails
F. 123 followed by an exception
G. 1234 followed by an exception
H. An exception is thrown with no other output
Answer: H
Nitin Manjunath


14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
4. public static void main(String[] args) {
5. foreach:
6. for(int j=0; j<5; j++) {// j = 0,1,2,3
7. for(int k=0; k< 3; k++) {// k = 0,1
8. System.out.print(" " + j);// 0 1 1 1 2 3 3
9. if(j==3 && k==1) break foreach;
10. if(j==0 || j==2) break;
11. }
12. }
13. }
14. }
What is the result?
A. 0 1 2 3
B. 1 1 1 3 3
C. 0 1 1 1 2 3 3
D. 1 1 1 3 3 4 4 4
E. 0 1 1 1 2 3 3 4 4 4
F. Compilation fails
Answer: C
Nitin Manjunath
Ln 80, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Mostly clear
Search
ENG
IN
22:30
14-12-2022


14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
3. public class Gotcha {
4. public static void main(String[] args) {
5. // insert code here
6.
7. }
8. void go() {
9. go();
10. }
11. }
And given the following three code fragments:
I. new Gotcha().go();
II. try { new Gotcha().go(); }
catch (Error e) { System.out.println("ouch"); }
III. try { new Gotcha().go(); }
catch (Exception e) { System.out.println("ouch"); }
When fragments I - III are added, independently, at line 5, which are true? (Choose all that apply.)
A. Some will not compile
B. They will all compile
C. All will complete normally
D. None will complete normally
E. Only one will complete normally
F. Two of them will complete normally
Answer: B,E
Nitin Manjunath
Ln 107, Col 1
80%
Windows (CRLF)
UTF-8
21°C
Mostly clear
Search
ENG
IN
22:39
14-12-2022


14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
Given:
1. public class Frisbee {
2. // insert code here
3. int x = 0;
4. System.out.println(7/x);
5. }
6. }
And given the following four code fragments:
I. public static void main(String[] args) {
II. public static void main(String[] args) throws Exception {
III. public static void main(String[] args) throws IOException {
IV. public static void main(String[] args) throws RuntimeException {
If the four fragments are inserted independently at line 2, which are true? (Choose all that apply.)
A. All four will compile and execute without exception
B. All four will compile and execute and throw an exception
C. Some, but not all, will compile and execute without exception
D. Some, but not all, will compile and execute and throw an exception
E. When considering fragments II, III, and IV, of those that will compile, adding a try/catch block around line 6 will cause compilation to fail.
i,ii,iv will compile and at the runtime throws ArithmeticException.
iii code only won't compile.
Answer:: D
Ln 134, Col 11
60%
Windows (CRLF)
UTF-8
Nitin Manjunath
20°C
Partly cloudy
Search
ENG
IN
22:49
14-12-2022


14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
i,ii,iv will compile and at the runtime throws ArithmeticException.
iii code only won't compile.
Answer:: D
Given:
2. class MyException extends Exception { }//MyException is a checkedException(partially checked one) only.
3. class Tire {
4. void doStuff() { }
5. }
6. public class Retread extends Tire {
7. public static void main(String[] args) {
8. new Retread().doStuff();
9. }
10. // insert code here
11. System.out.println(7/0);//ArithmeticException(uncheckedException)
12. }
13. }
And given the following four code fragments:
I. void doStuff() {
II. void doStuff() throws MyException {
III. void doStuff() throws RuntimeException {
IV. void doStuff() throws ArithmeticException {
When fragments I - IV are added, independently, at line 10, which are true? (Choose all that apply.)
A. None will compile
B. They will all compile
Nitin Manjunath
Ln 168, Col 12
80%
Windows (CRLF)
UTF-8
20°C
Partly cloudy
Search
ENG
IN
22:55
14-12-2022


14_12_2022_Snippets_sessions_notes - Notepad
File Edit View
And given the following four code fragments:
I. void doStuff() {
II. void doStuff() throws MyException {
III. void doStuff() throws RuntimeException {
IV. void doStuff() throws ArithmeticException {
When fragments I - IV are added, independently, at line 10, which are true? (Choose all that apply.)
A. None will compile
B. They will all compile
C. Some, but not all, will compile
D. All of those that compile will throw an exception at runtime
E. None of those that compile will throw an exception at runtime
F. Only some of those that compile will throw an exception at runtime
i valid
ii CE
iii valid
iv valid
Answer: C,D
Nitin Manjunath
Ln 161, Col 36
60%
Windows (CRLF)
UTF-8
20°C
Partly cloudy
Search
ENG
IN
22:56
14-12-2022