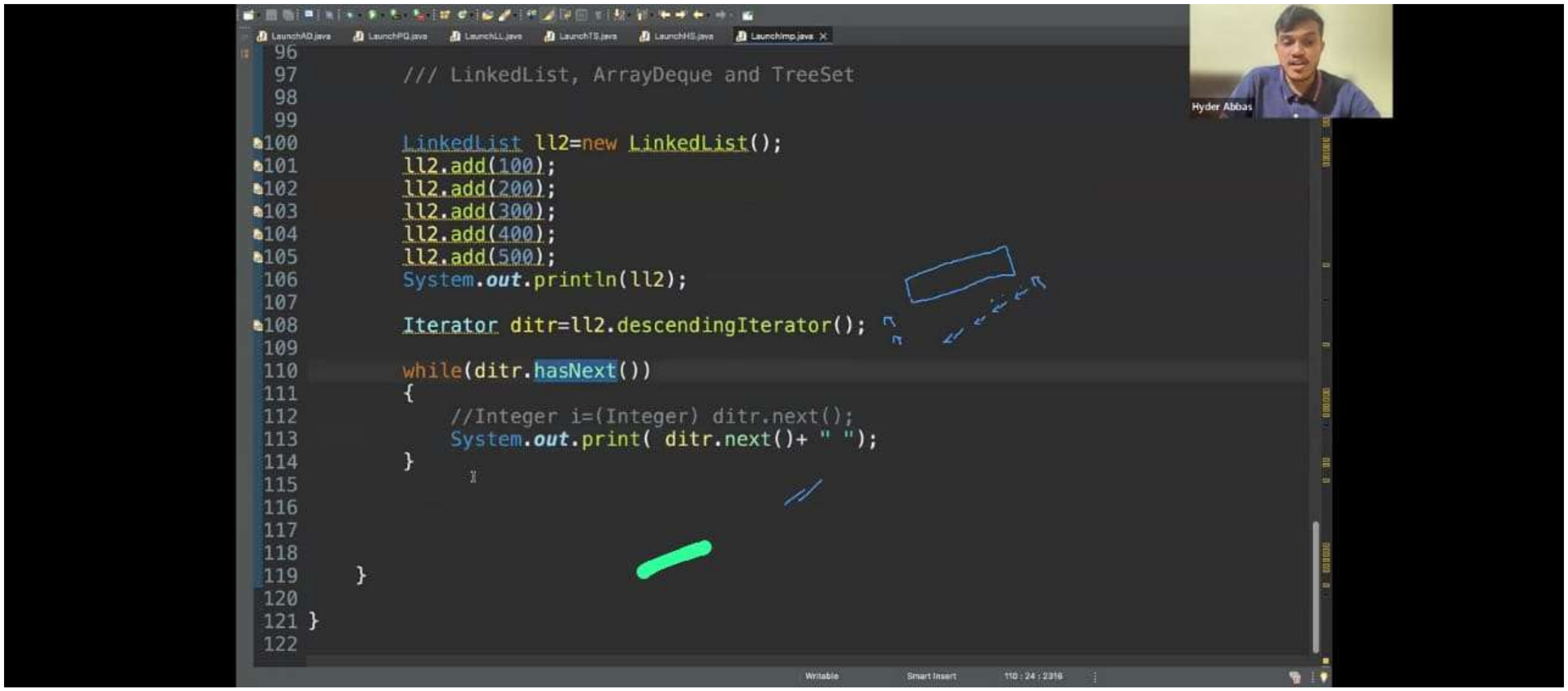
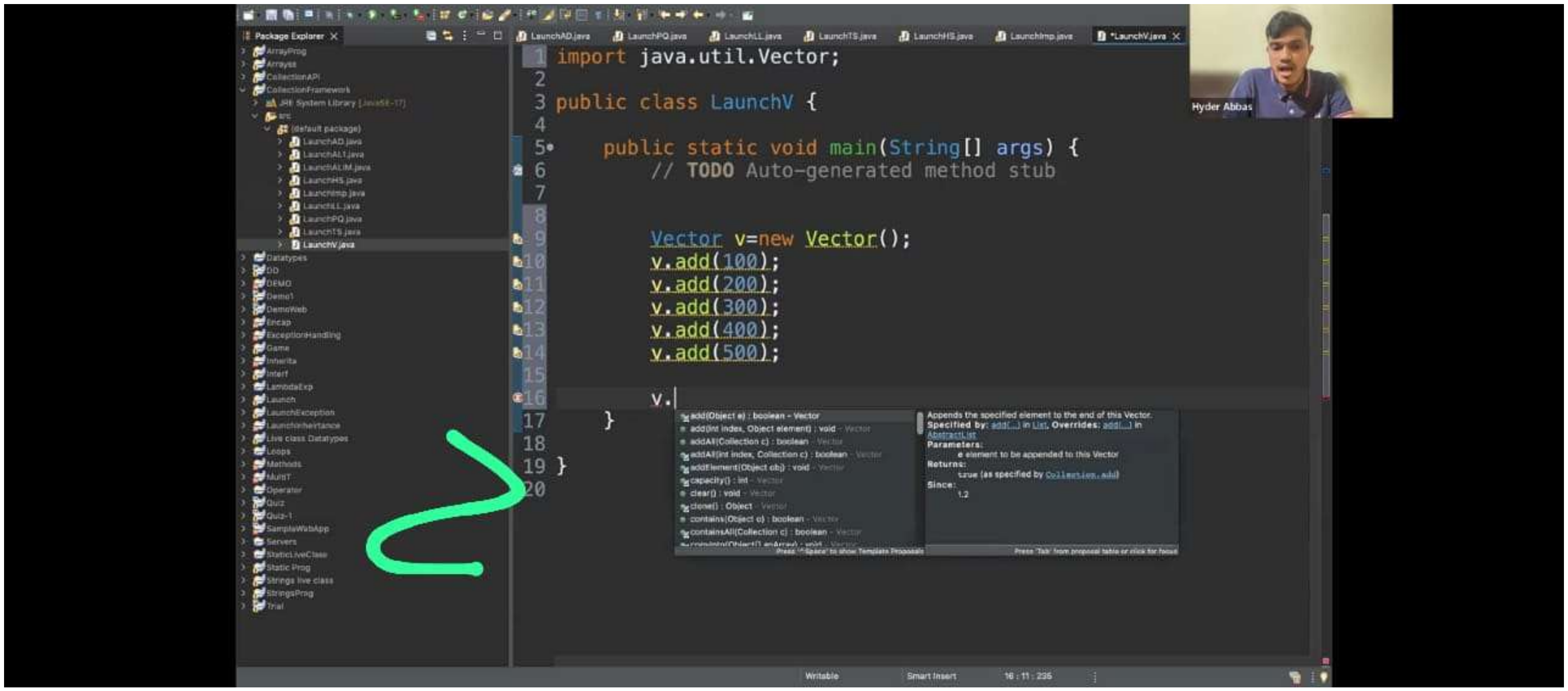
Java Collection Part-3



```java
        /// LinkedList, ArrayDeque and TreeSet


        LinkedList ll2=new LinkedList();
        ll2.add(100);
        ll2.add(200);
        ll2.add(300);
        ll2.add(400);
        ll2.add(500);
        System.out.println(ll2);

        Iterator ditr=ll2.descendingIterator();

        while(ditr.hasNext())
        {
            //Integer i=(Integer) ditr.next();
            System.out.print( ditr.next()+ " ");
        }





    }

}
```

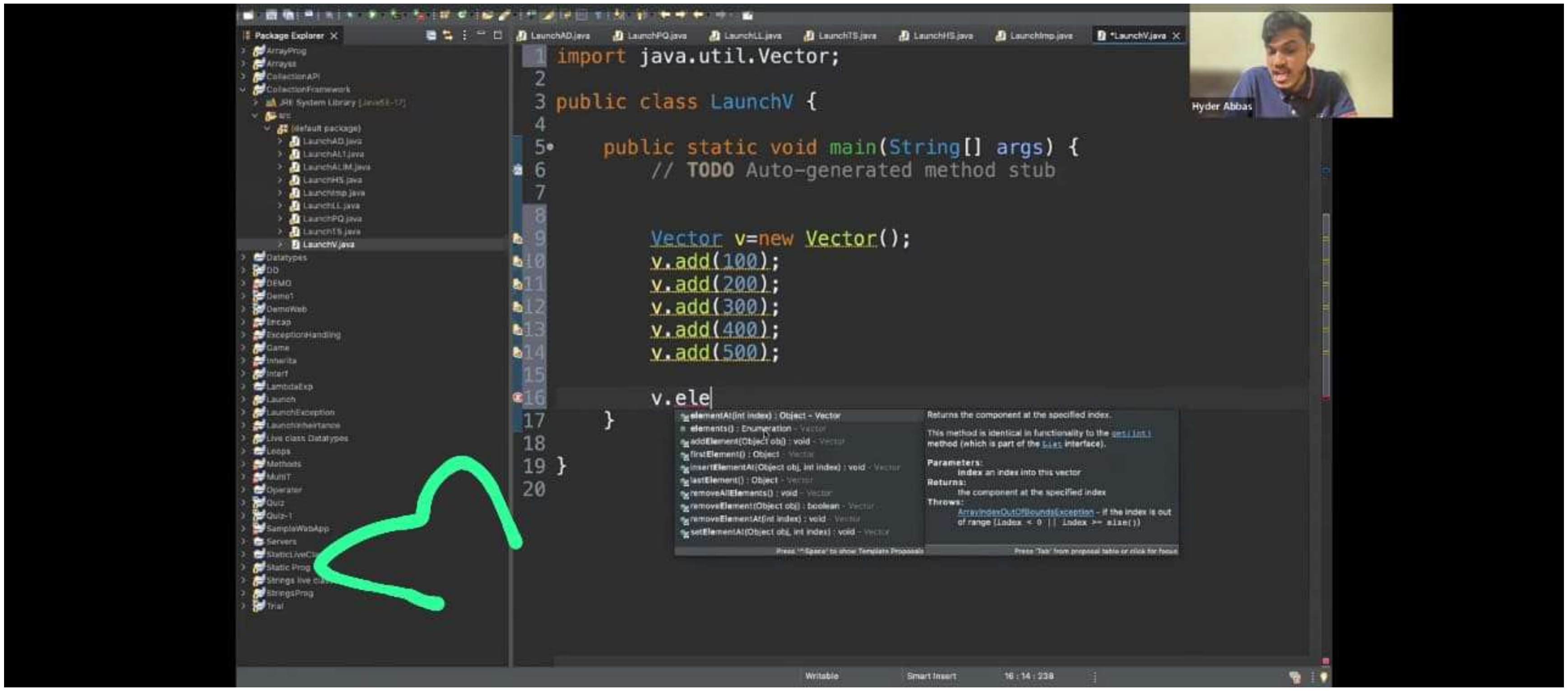


```java
import java.util.Vector;

public class LaunchV {

    public static void main(String[] args) {
        // TODO Auto-generated method stub


        Vector v=new Vector();
        v.add(100);
        v.add(200);
        v.add(300);
        v.add(400);
        v.add(500);

        v.
    }
}
```

```
add(Object e) : boolean - Vector
add(int index, Object element) : void - Vector
addAll(Collection c) : boolean - Vector
addAll(int index, Collection c) : boolean - Vector
addElement(Object obj) : void - Vector
capacity() : int - Vector
clear() : void - Vector
clone() : Object - Vector
contains(Object o) : boolean - Vector
containsAll(Collection c) : boolean - Vector
Press '^Space' to show Template Proposals
```

Appends the specified element to the end of this Vector.
Specified by: add(...) in List. **Overrides:** add(...) in AbstractList
Parameters:
e element to be appended to this Vector
Returns:
true (as specified by Collection.add)
Since:
1.2

Press 'Tab' from proposal table or click for focus



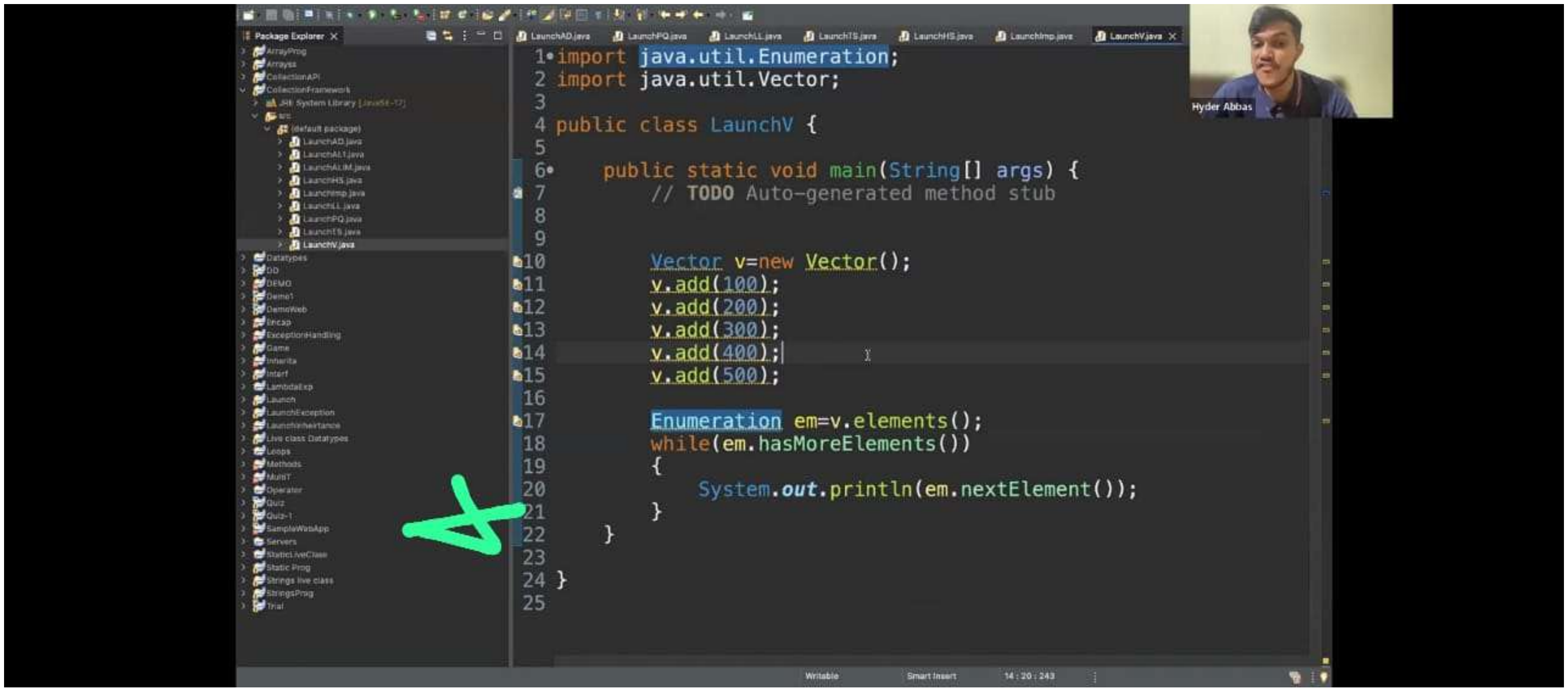
```java
import java.util.Vector;

public class LaunchV {

    public static void main(String[] args) {
        // TODO Auto-generated method stub


        Vector v=new Vector();
        v.add(100);
        v.add(200);
        v.add(300);
        v.add(400);
        v.add(500);

        v.ele
    }

}
```

```
elementAt(int index) : Object - Vector
elements() : Enumeration - Vector
addElement(Object obj) : void - Vector
firstElement() : Object - Vector
insertElementAt(Object obj, int index) : void - Vector
lastElement() : Object - Vector
removeAllElements() : void - Vector
removeElement(Object obj) : boolean - Vector
removeElementAt(int index) : void - Vector
setElementAt(Object obj, int index) : void - Vector
```

Returns the component at the specified index.

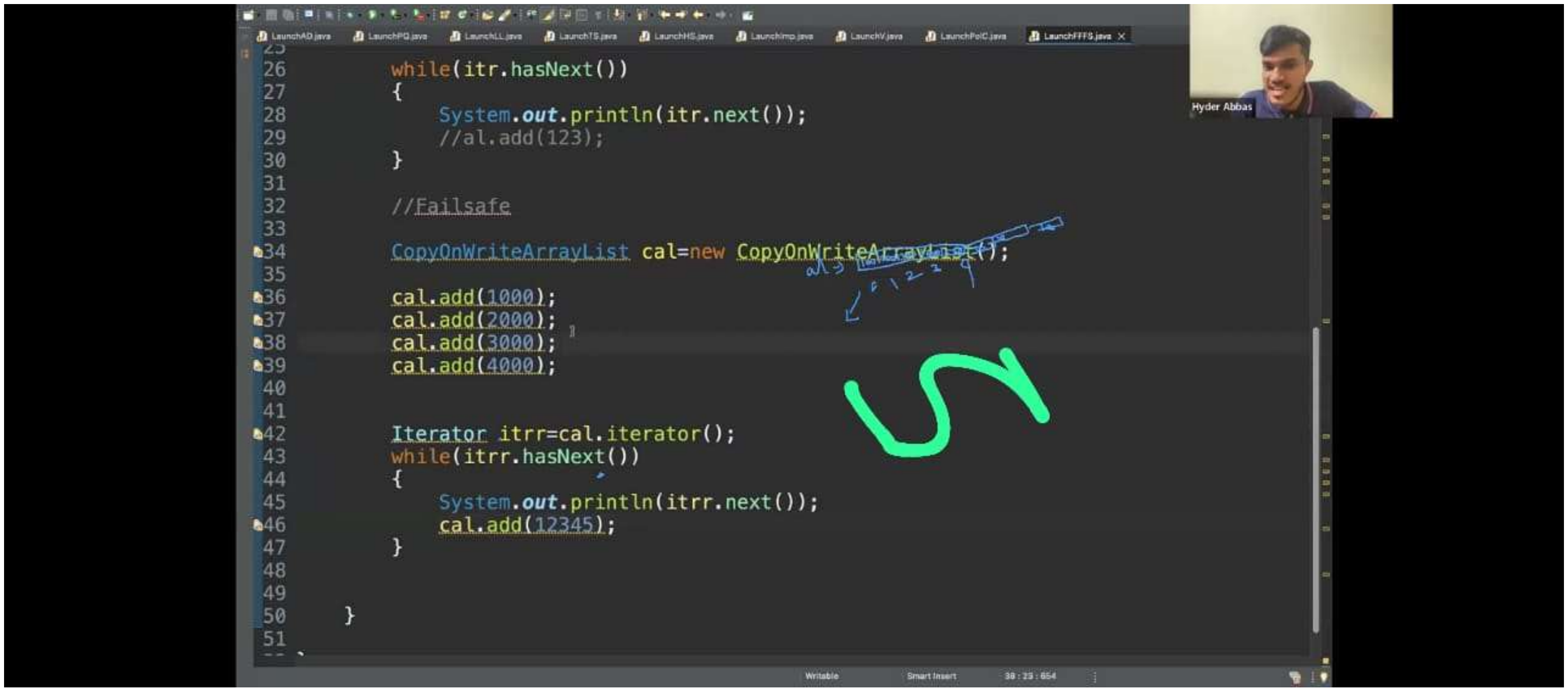
This method is identical in functionality to the get(int) method (which is part of the List interface).

Parameters:
    index an index into this vector
Returns:
    the component at the specified index
Throws:
    ArrayIndexOutOfBoundsException - if the index is out of range (index < 0 || index >= size())

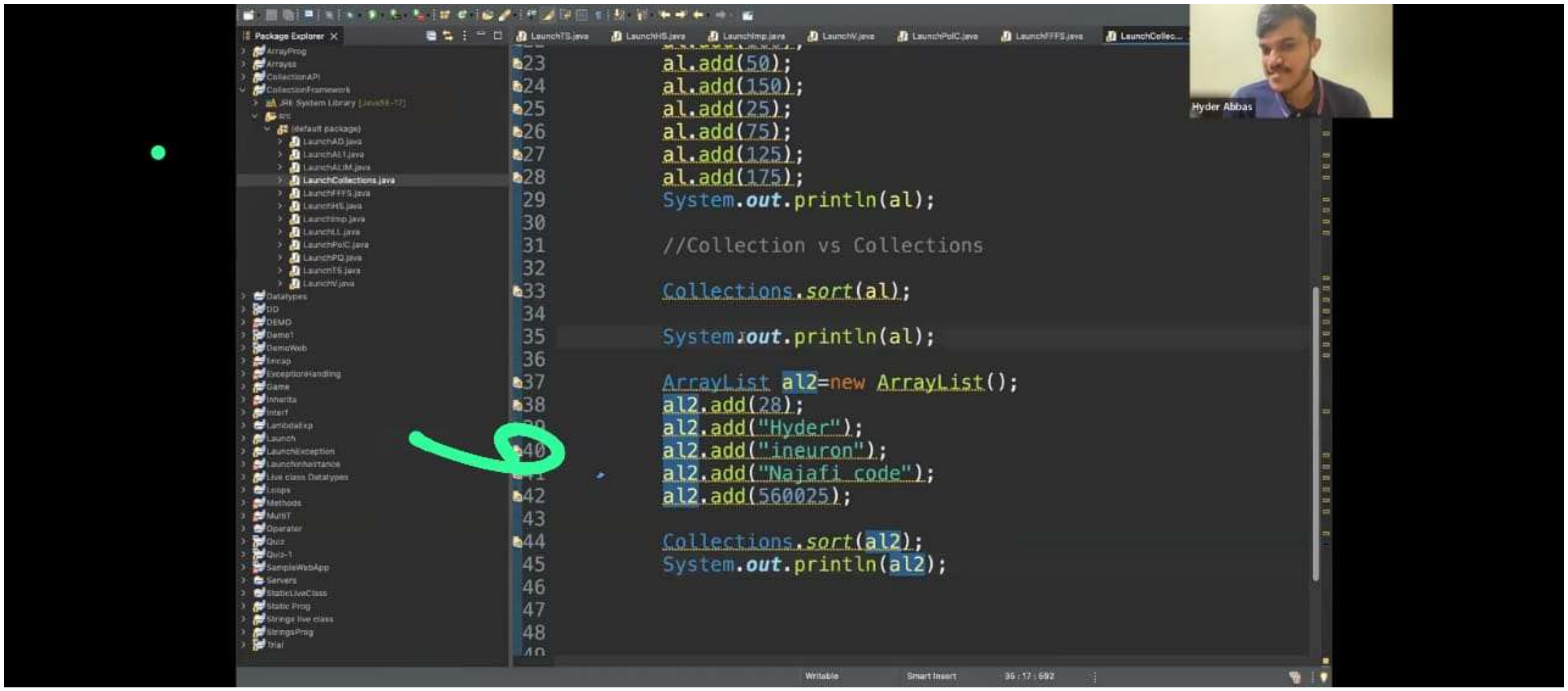


```java
import java.util.Enumeration;
import java.util.Vector;

public class LaunchV {

    public static void main(String[] args) {
        // TODO Auto-generated method stub


        Vector v=new Vector();
        v.add(100);
        v.add(200);
        v.add(300);
        v.add(400);
        v.add(500);

        Enumeration em=v.elements();
        while(em.hasMoreElements())
        {
            System.out.println(em.nextElement());
        }
    }

}
```

4

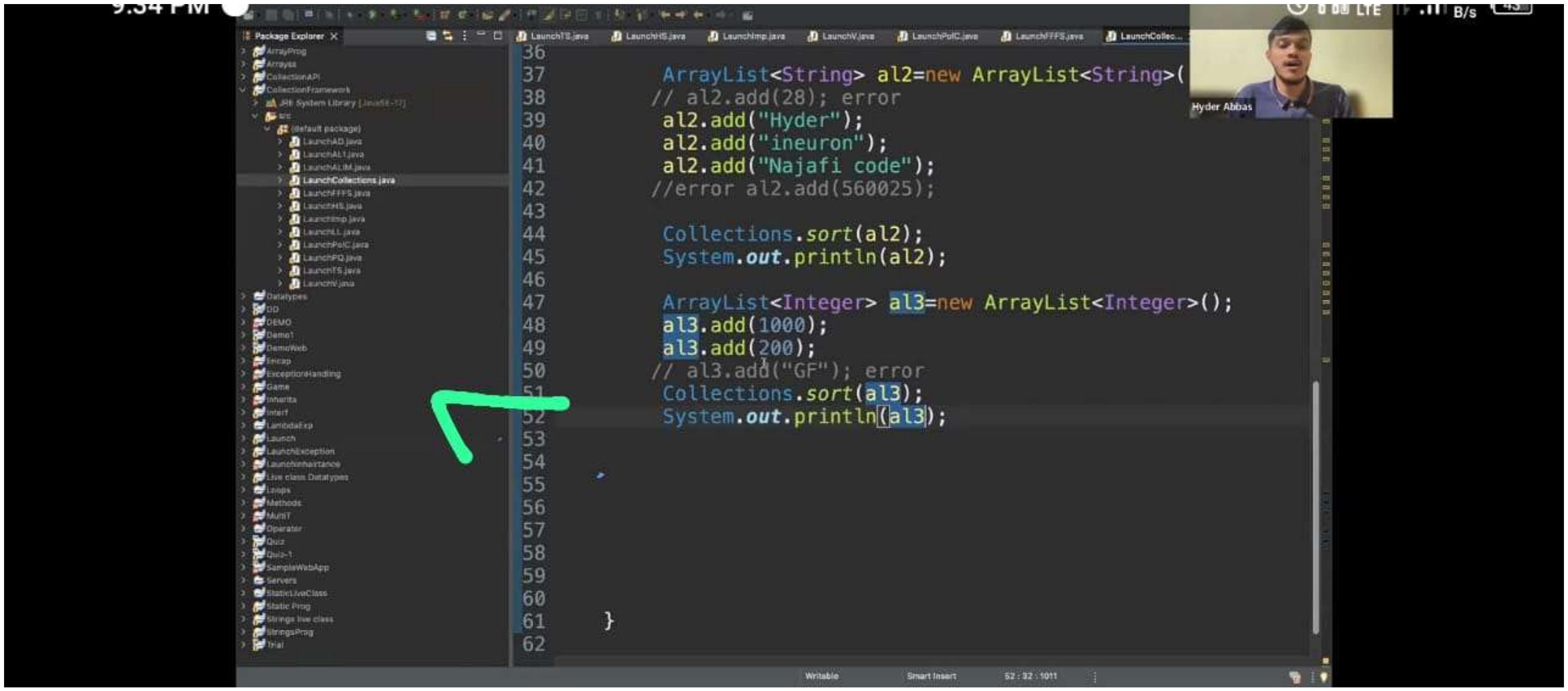


```java
        while(itr.hasNext())
        {
            System.out.println(itr.next());
            //al.add(123);
        }

        //Failsafe

        CopyOnWriteArrayList cal=new CopyOnWriteArrayList();

        cal.add(1000);
        cal.add(2000);
        cal.add(3000);
        cal.add(4000);


        Iterator itrr=cal.iterator();
        while(itrr.hasNext())
        {
            System.out.println(itrr.next());
            cal.add(12345);
        }


    }
```



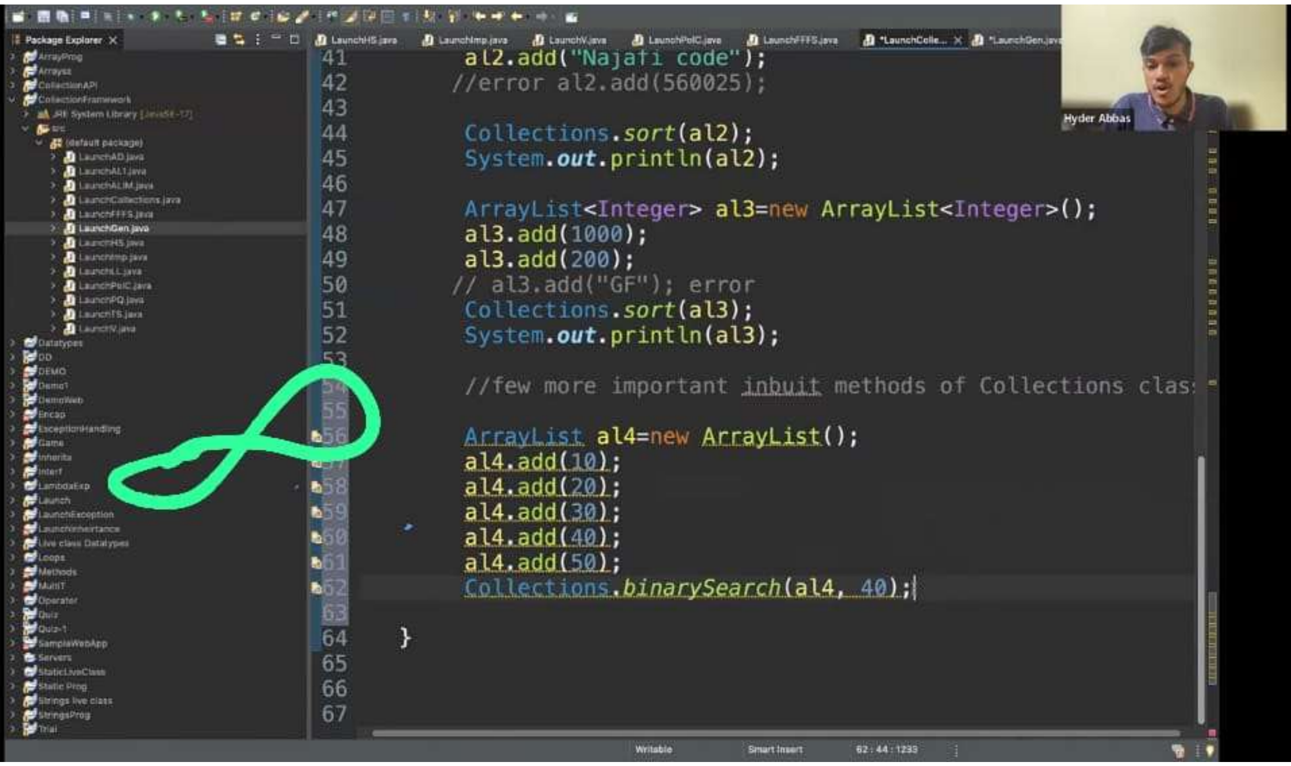
```java
        al.add(50);
        al.add(150);
        al.add(25);
        al.add(75);
        al.add(125);
        al.add(175);
        System.out.println(al);

        //Collection vs Collections

        Collections.sort(al);

        System.out.println(al);

        ArrayList al2=new ArrayList();
        al2.add(28);
        al2.add("Hyder");
        al2.add("ineuron");
        al2.add("Najafi code");
        al2.add(560025);

        Collections.sort(al2);
        System.out.println(al2);
```

6

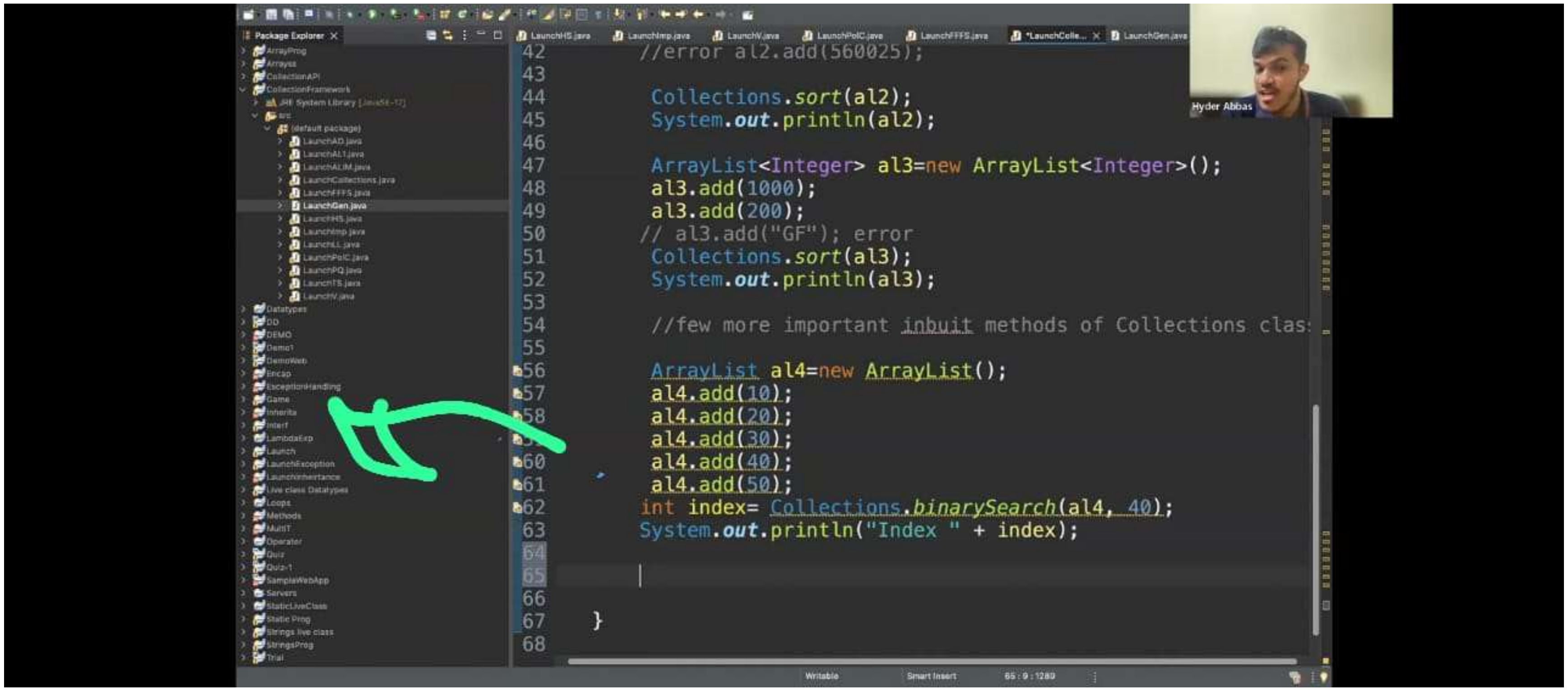


7

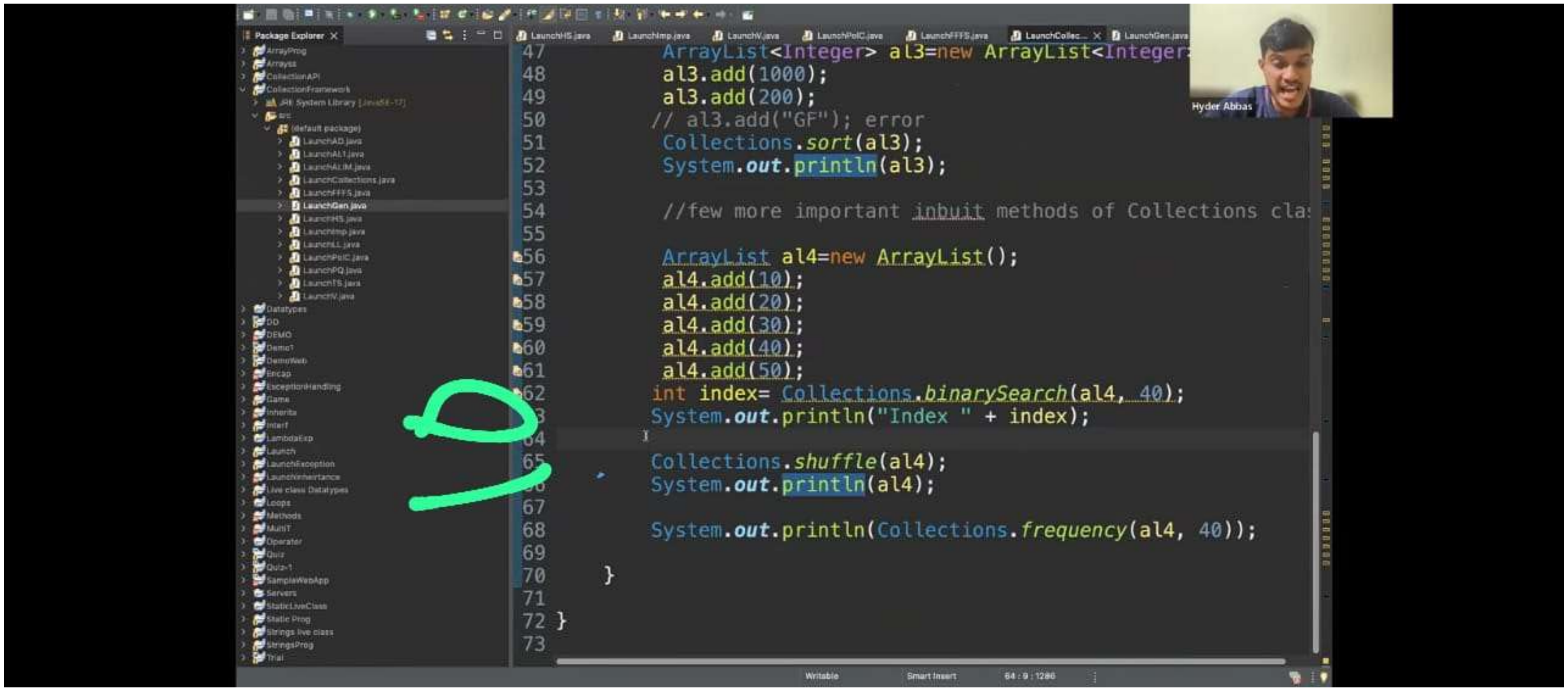
```java
            ArrayList<String> al2=new ArrayList<String>(
          // al2.add(28); error
            al2.add("Hyder");
            al2.add("ineuron");
            al2.add("Najafi code");
          //error al2.add(560025);

            Collections.sort(al2);
            System.out.println(al2);

            ArrayList<Integer> al3=new ArrayList<Integer>();
            al3.add(1000);
            al3.add(200);
          // al3.add("GF"); error
            Collections.sort(al3);
            System.out.println(al3);








        }

```



```java
        al2.add("Najafi code");
       //error al2.add(560025);

        Collections.sort(al2);
        System.out.println(al2);

        ArrayList<Integer> al3=new ArrayList<Integer>();
        al3.add(1000);
        al3.add(200);
       // al3.add("GF"); error
        Collections.sort(al3);
        System.out.println(al3);

        //few more important inbuit methods of Collections clas

        ArrayList al4=new ArrayList();
        al4.add(10);
        al4.add(20);
        al4.add(30);
        al4.add(40);
        al4.add(50);
        Collections.binarySearch(al4, 40);

    }
```

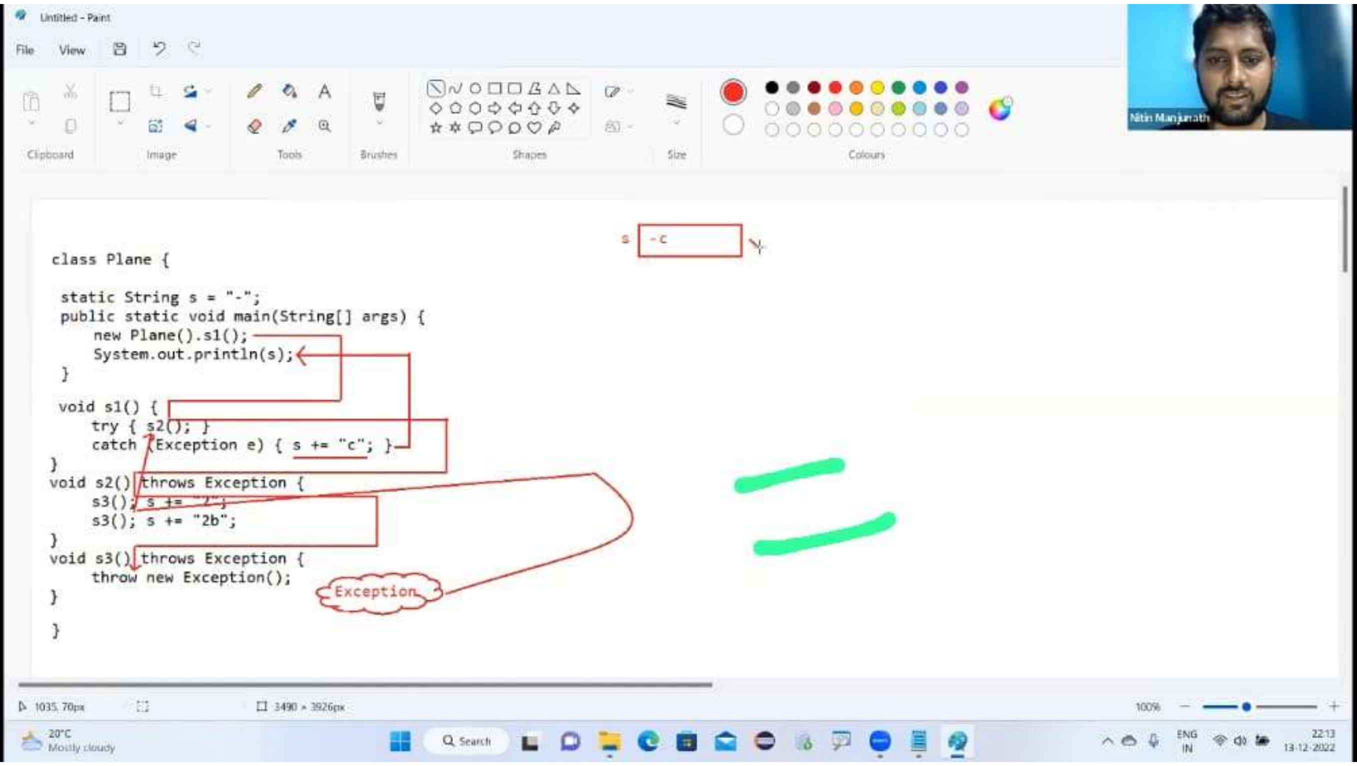


```java
        //error al2.add(560025);

         Collections.sort(al2);
         System.out.println(al2);

         ArrayList<Integer> al3=new ArrayList<Integer>();
         al3.add(1000);
         al3.add(200);
        // al3.add("GF"); error
         Collections.sort(al3);
         System.out.println(al3);

         //few more important inbuit methods of Collections clas

         ArrayList al4=new ArrayList();
         al4.add(10);
         al4.add(20);
         al4.add(30);
         al4.add(40);
         al4.add(50);
        int index= Collections.binarySearch(al4, 40);
        System.out.println("Index " + index);


    }

```

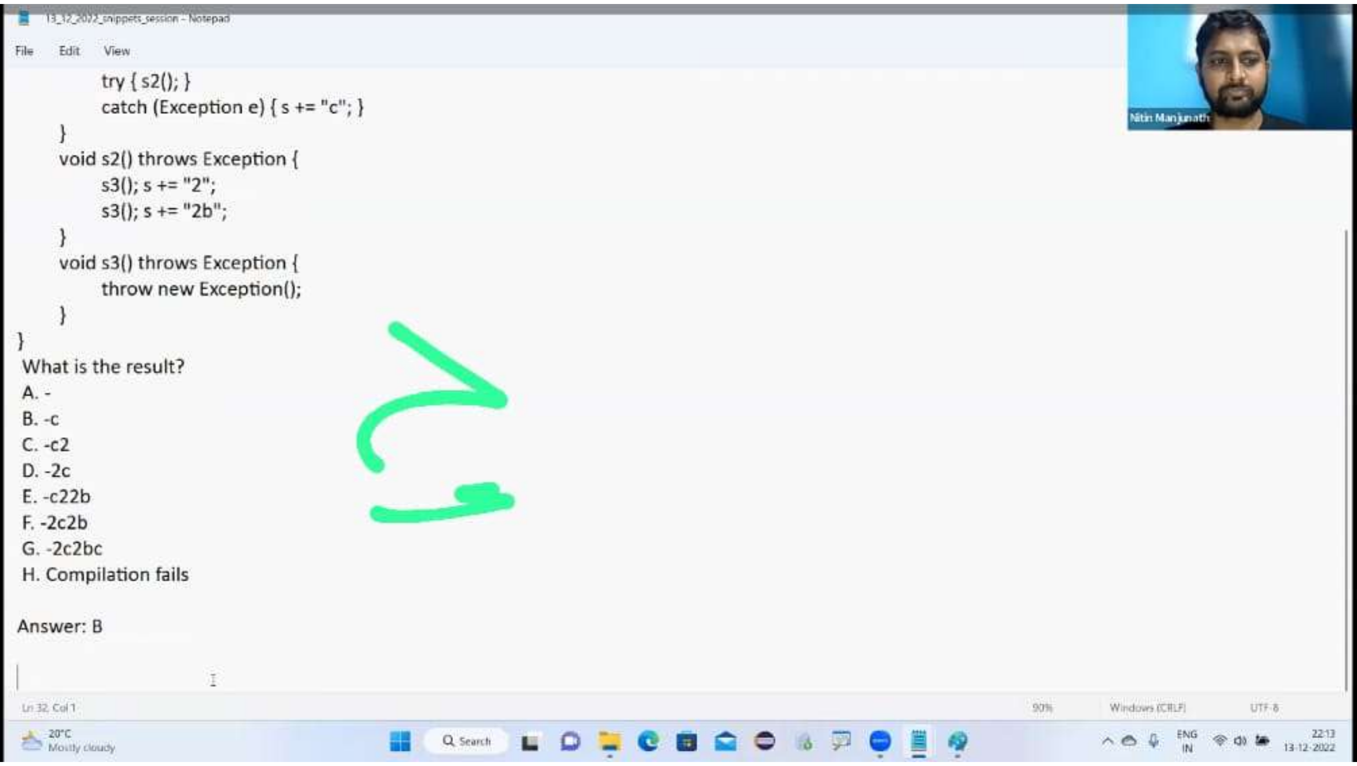


```java
        ArrayList<Integer> al3=new ArrayList<Integer
        al3.add(1000);
        al3.add(200);
       // al3.add("GF"); error
        Collections.sort(al3);
        System.out.println(al3);

        //few more important inbuit methods of Collections cla

        ArrayList al4=new ArrayList();
        al4.add(10);
        al4.add(20);
        al4.add(30);
        al4.add(40);
        al4.add(50);
       int index= Collections.binarySearch(al4, 40);
       System.out.println("Index " + index);

       Collections.shuffle(al4);
       System.out.println(al4);

       System.out.println(Collections.frequency(al4, 40));

    }

}
```

10



```
class Plane {

 static String s = "-";
 public static void main(String[] args) {
     new Plane().s1();
     System.out.println(s);
 }

 void s1() {
     try { s2(); }
     catch (Exception e) { s += "c"; }
}
void s2() throws Exception {
     s3(); s += "2";
     s3(); s += "2b";
}
void s3() throws Exception {
     throw new Exception();
}

}
```

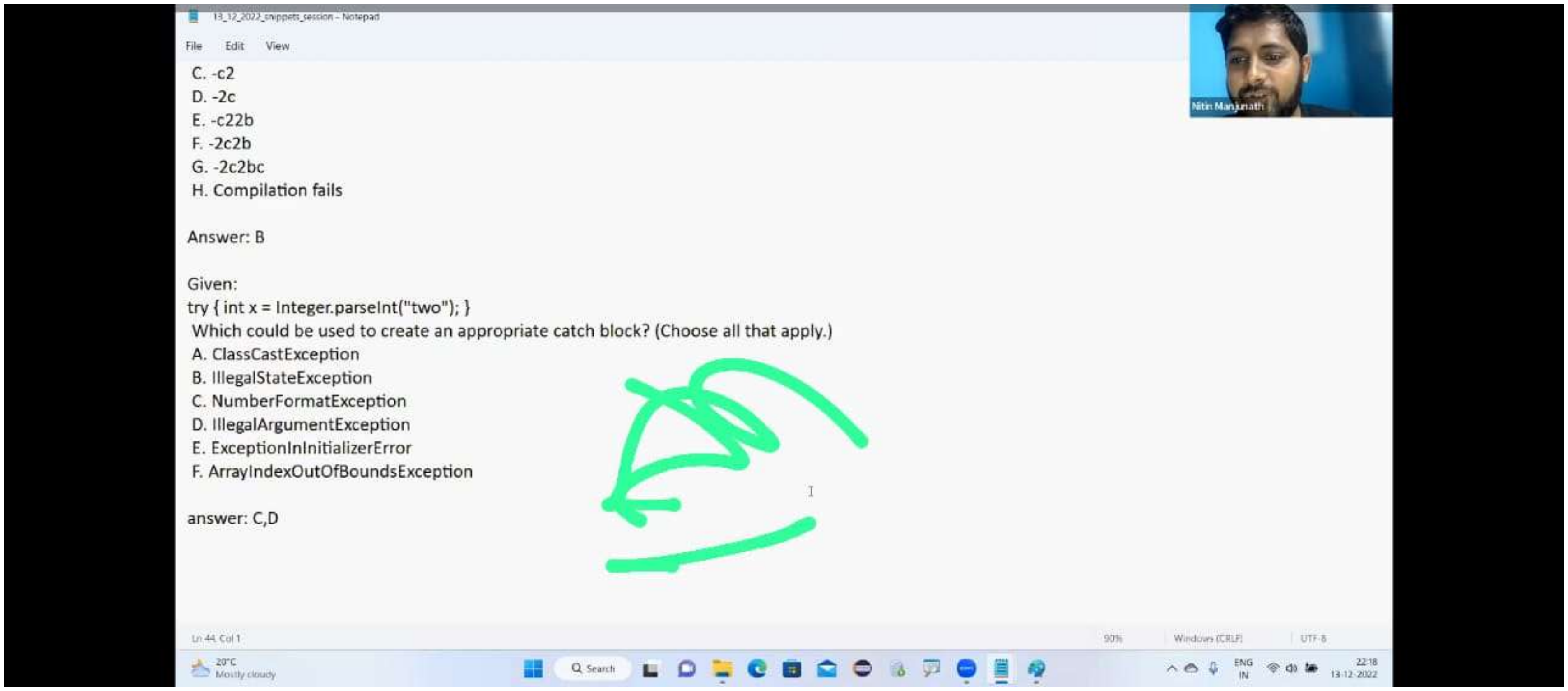


```
            try { s2(); }
            catch (Exception e) { s += "c"; }
      }
      void s2() throws Exception {
            s3(); s += "2";
            s3(); s += "2b";
      }
      void s3() throws Exception {
            throw new Exception();
      }
}
```

What is the result?
A. -
B. -c
C. -c2
D. -2c
E. -c22b
F. -2c2b
G. -2c2bc
H. Compilation fails

Answer: B

12



C. -c2
D. -2c
E. -c22b
F. -2c2b
G. -2c2bc
H. Compilation fails

Answer: B

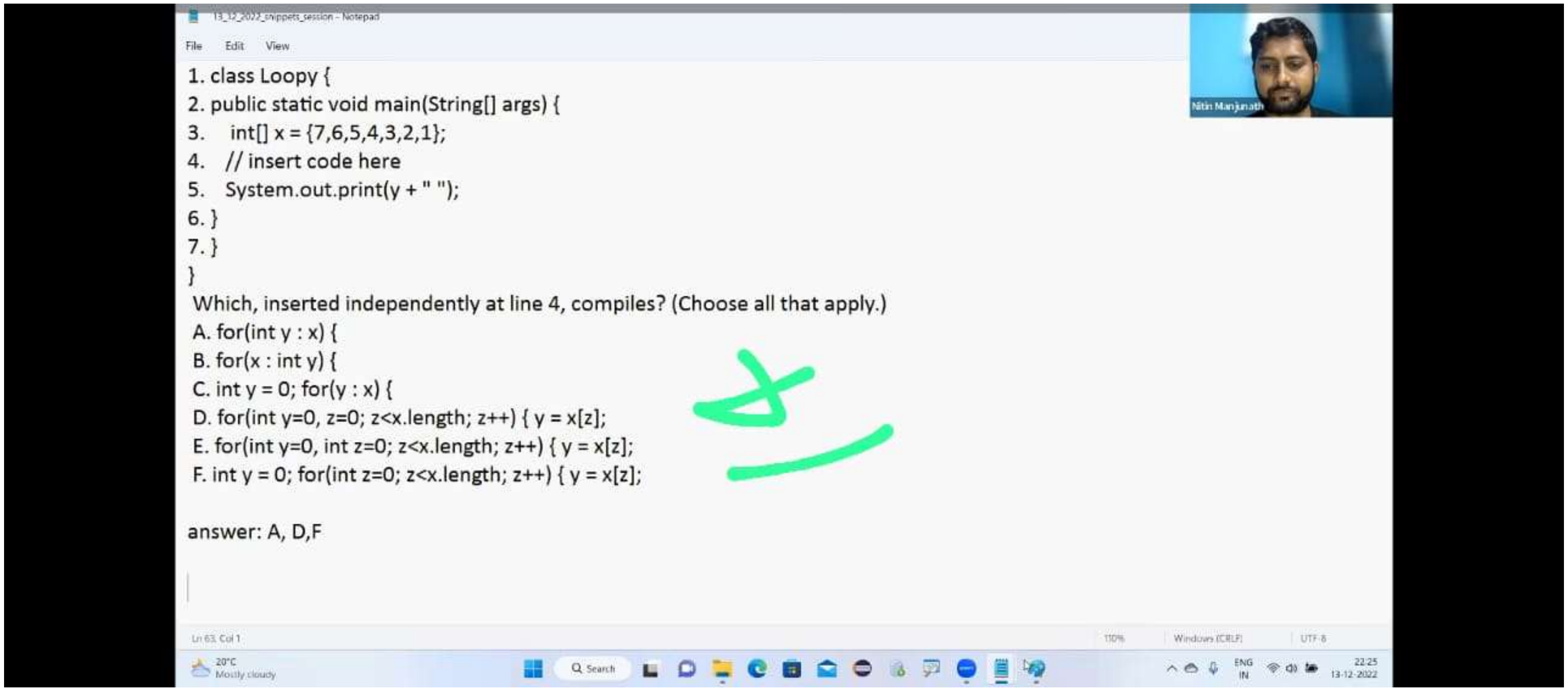
Given:

```
try { int x = Integer.parseInt("two"); }
```

Which could be used to create an appropriate catch block? (Choose all that apply.)

A. ClassCastException
B. IllegalStateException
C. NumberFormatException
D. IllegalArgumentException
E. ExceptionInInitializerError
F. ArrayIndexOutOfBoundsException

answer: C,D


13_12_2022_snippets_session - Notepad

File Edit View

```
1. class Loopy {
2. public static void main(String[] args) {
3.    int[] x = {7,6,5,4,3,2,1};
4.   // insert code here
5.   System.out.print(y + " ");
6. }
7. }
}
```

Which, inserted independently at line 4, compiles? (Choose all that apply.)

A. for(int y : x) {

B. for(x : int y) {

C. int y = 0; for(y : x) {

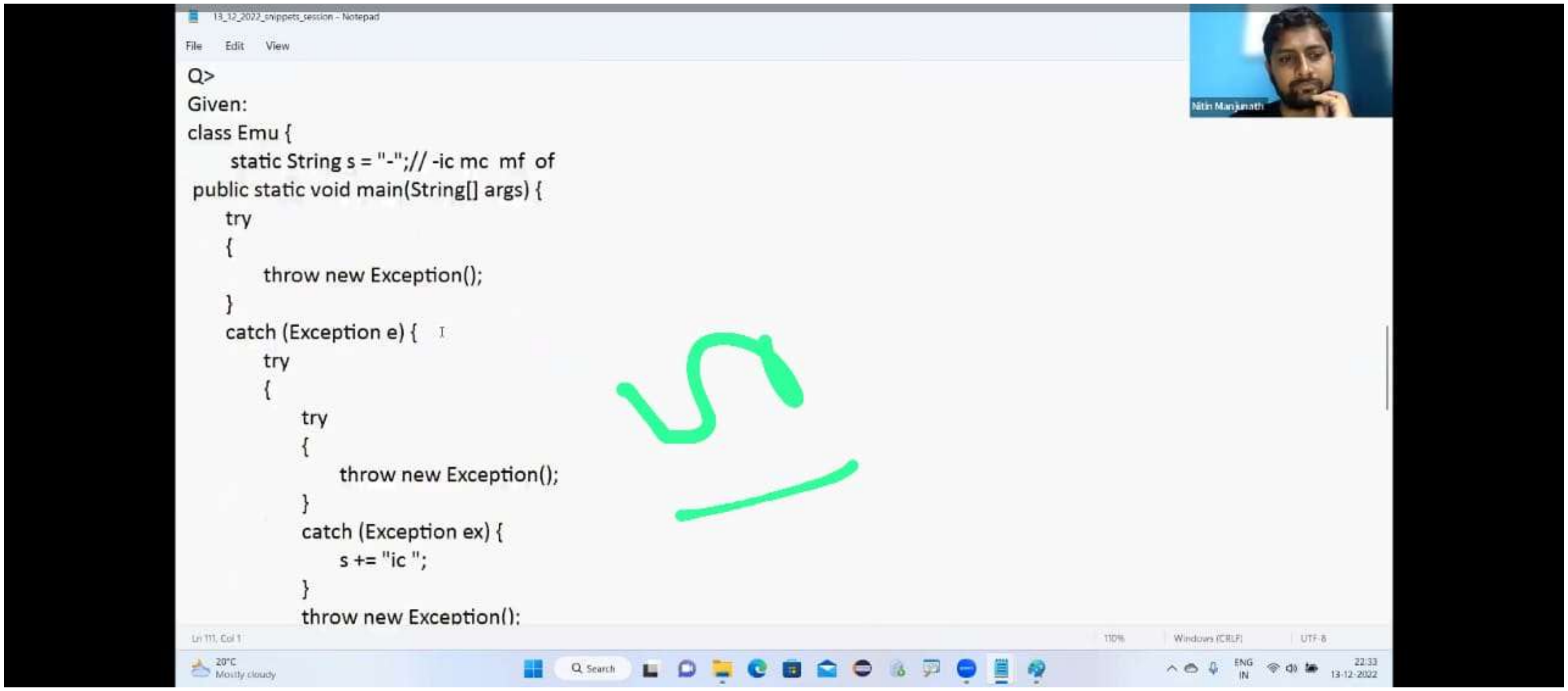
D. for(int y=0, z=0; z<x.length; z++) { y = x[z];

E. for(int y=0, int z=0; z<x.length; z++) { y = x[z];

F. int y = 0; for(int z=0; z<x.length; z++) { y = x[z];

answer: A, D,F

14

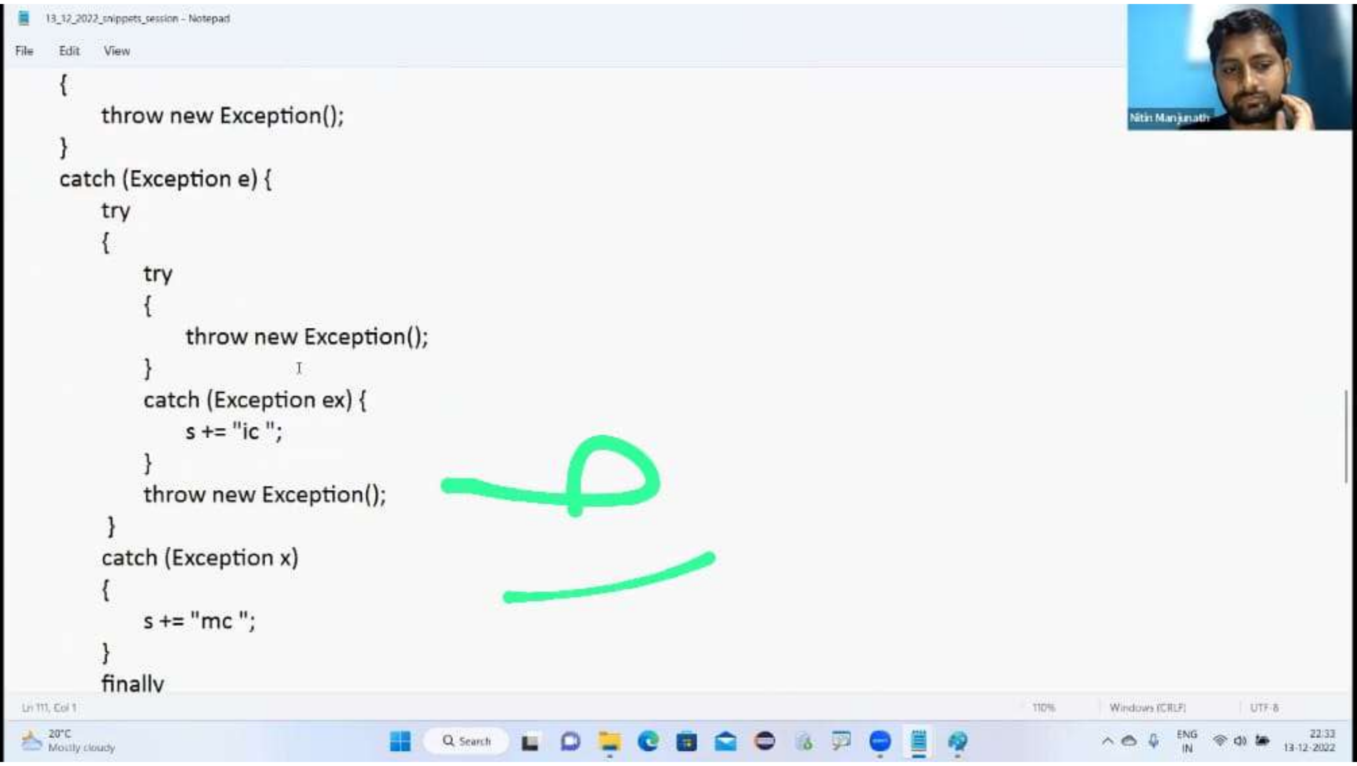


13_12_2022_snippets_session - Notepad

File Edit View

```
Q>
Given:
class Emu {
     static String s = "-";// -ic mc  mf  of
 public static void main(String[] args) {
    try
    {
        throw new Exception();
    }
    catch (Exception e) {
        try
        {
            try
            {
                throw new Exception();
            }
            catch (Exception ex) {
                s += "ic ";
            }
            throw new Exception();
```

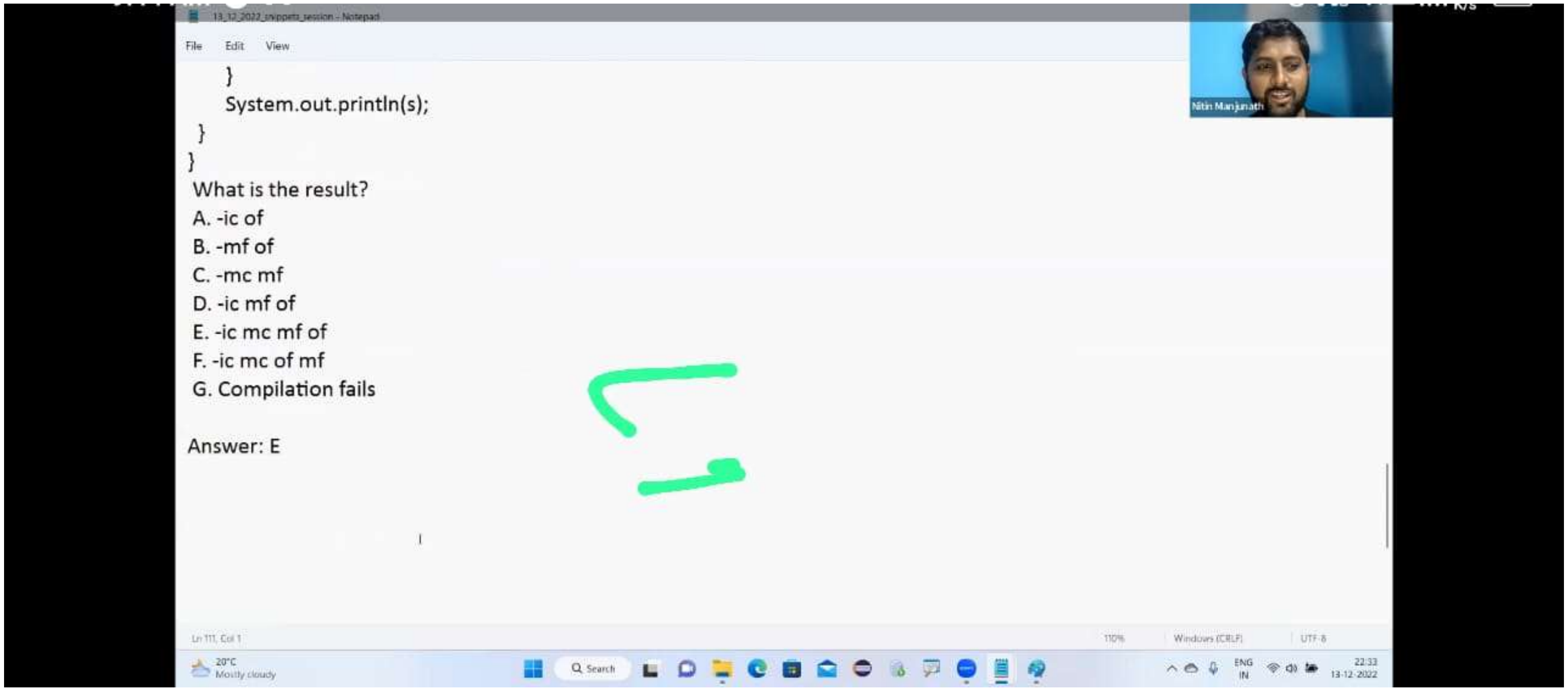
15



13_12_2022_snippets_session - Notepad

```
{
    throw new Exception();
}
catch (Exception e) {
    try
    {
        try
        {
            throw new Exception();
        }
        catch (Exception ex) {
            s += "ic ";
        }
        throw new Exception();
     }
    catch (Exception x)
    {
        s += "mc ";
    }
    finally
```

16

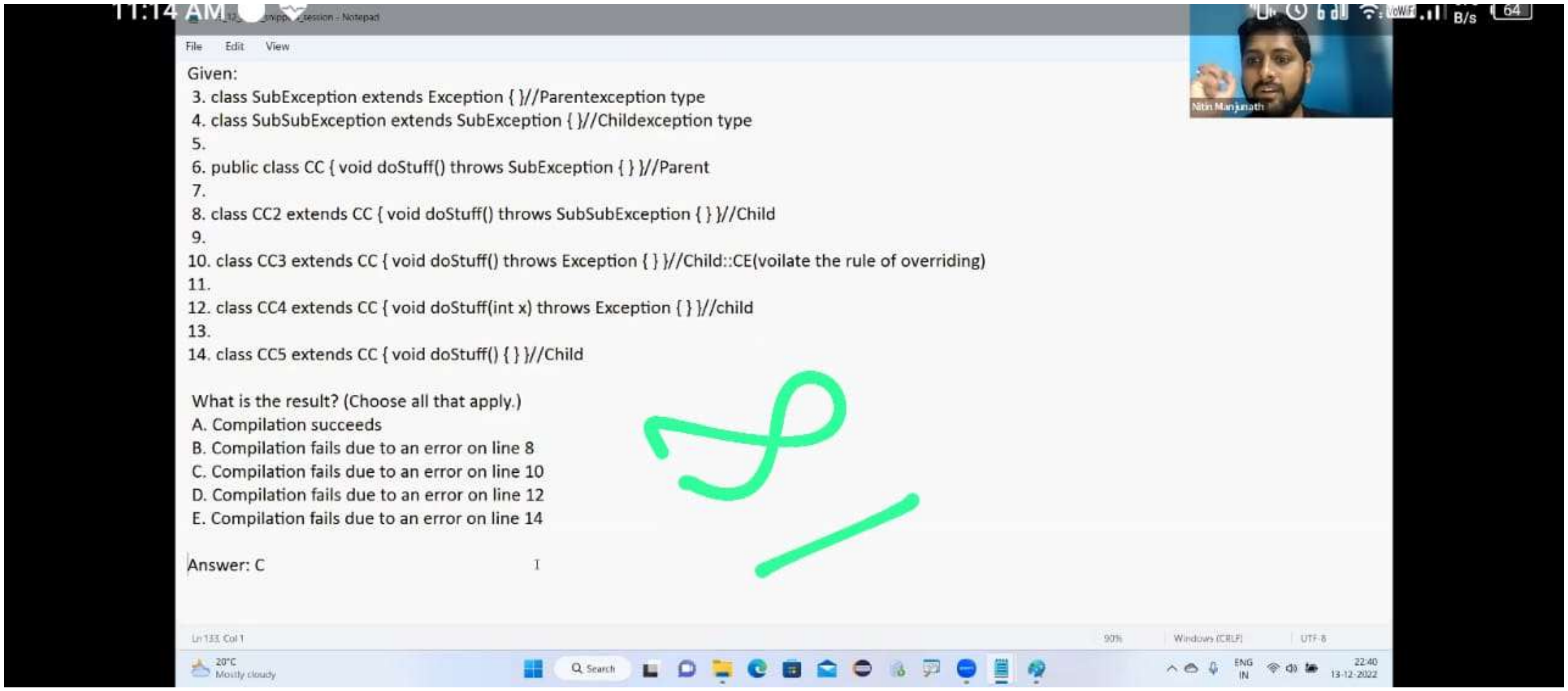


```
    }
    System.out.println(s);
  }
}
```

What is the result?
A. -ic of
B. -mf of
C. -mc mf
D. -ic mf of
E. -ic mc mf of
F. -ic mc of mf
G. Compilation fails

Answer: E

17



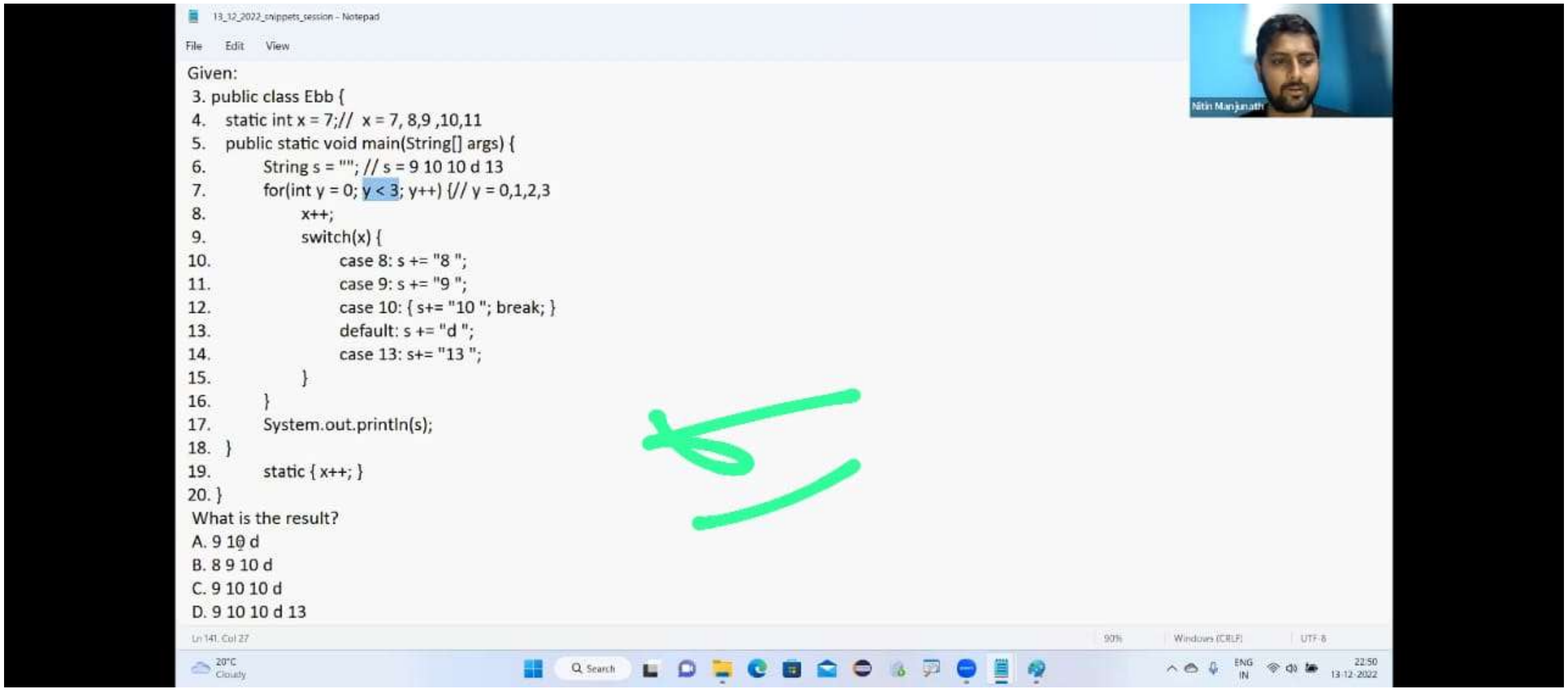
Given:

```
3. class SubException extends Exception { }//Parentexception type
4. class SubSubException extends SubException { }//Childexception type
5.
6. public class CC { void doStuff() throws SubException { } }//Parent
7.
8. class CC2 extends CC { void doStuff() throws SubSubException { } }//Child
9.
10. class CC3 extends CC { void doStuff() throws Exception { } }//Child::CE(voilate the rule of overriding)
11.
12. class CC4 extends CC { void doStuff(int x) throws Exception { } }//child
13.
14. class CC5 extends CC { void doStuff() { } }//Child
```

What is the result? (Choose all that apply.)
A. Compilation succeeds
B. Compilation fails due to an error on line 8
C. Compilation fails due to an error on line 10
D. Compilation fails due to an error on line 12
E. Compilation fails due to an error on line 14

Answer: C

18



```
Given:
 3. public class Ebb {
 4.   static int x = 7;//  x = 7, 8,9 ,10,11
 5.   public static void main(String[] args) {
 6.         String s = ""; // s = 9 10 10 d 13
 7.         for(int y = 0; y < 3; y++) {// y = 0,1,2,3
 8.               x++;
 9.               switch(x) {
10.                     case 8: s += "8 ";
11.                     case 9: s += "9 ";
12.                     case 10: { s+= "10 "; break; }
13.                     default: s += "d ";
14.                     case 13: s+= "13 ";
15.               }
16.         }
17.         System.out.println(s);
18.   }
19.         static { x++; }
20. }
```

What is the result?

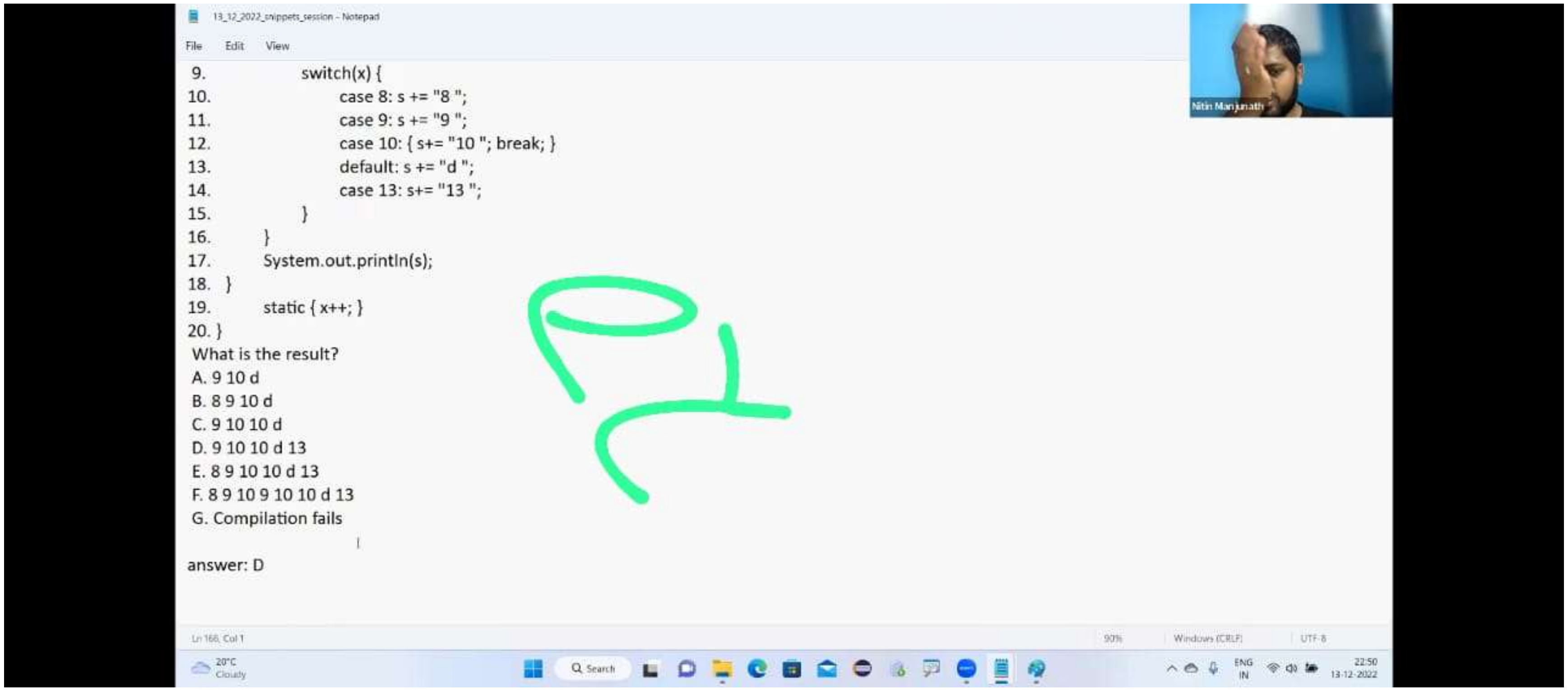
A. 9 10 d

B. 8 9 10 d

C. 9 10 10 d

D. 9 10 10 d 13

19

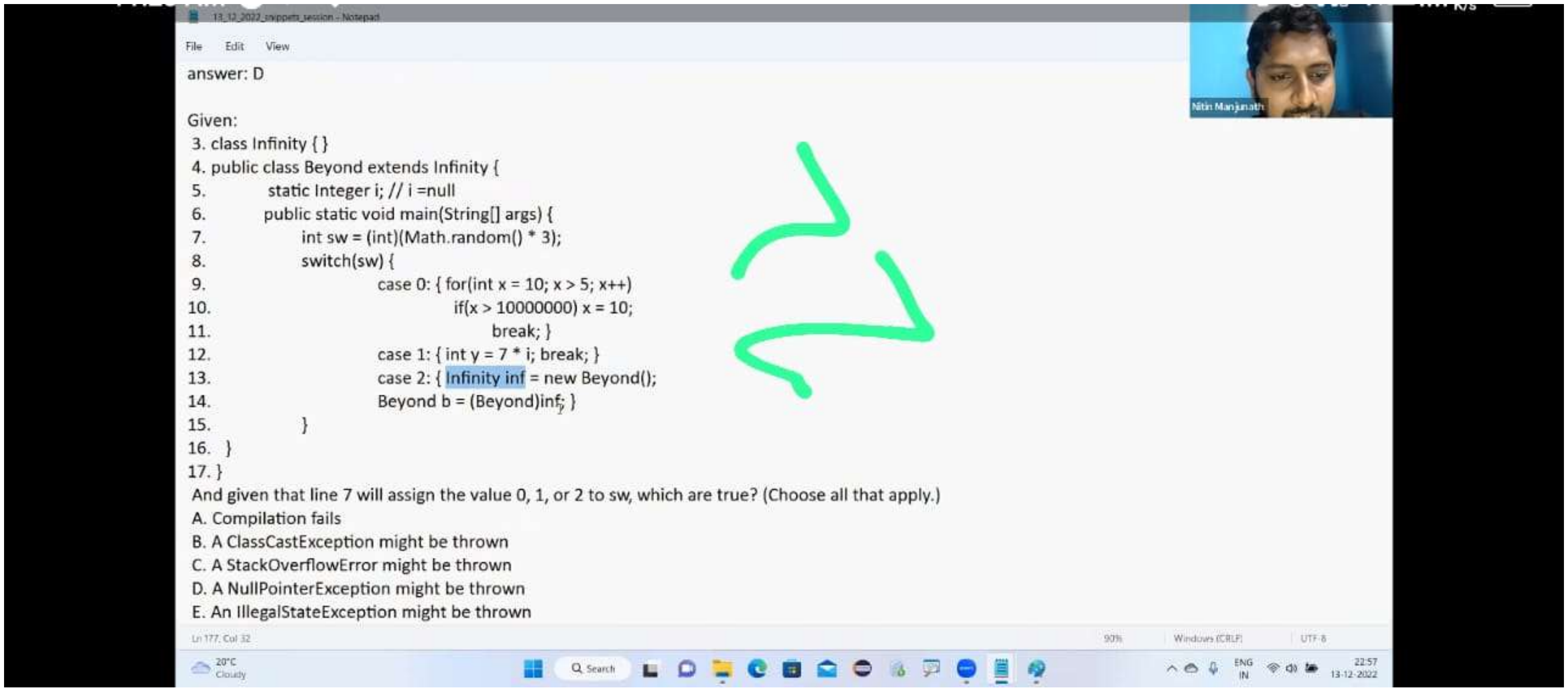


```
        switch(x) {
              case 8: s += "8 ";
              case 9: s += "9 ";
              case 10: { s+= "10 "; break; }
              default: s += "d ";
              case 13: s+= "13 ";
        }
    }
    System.out.println(s);
 }
    static { x++; }
}
```

What is the result?
A. 9 10 d
B. 8 9 10 d
C. 9 10 10 d
D. 9 10 10 d 13
E. 8 9 10 10 d 13
F. 8 9 10 9 10 10 d 13
G. Compilation fails

answer: D

20



answer: D

Given:

```
3. class Infinity { }
4. public class Beyond extends Infinity {
5.          static Integer i; // i =null
6.          public static void main(String[] args) {
7.                int sw = (int)(Math.random() * 3);
8.                switch(sw) {
9.                        case 0: { for(int x = 10; x > 5; x++)
10.                                   if(x > 10000000) x = 10;
11.                                       break; }
12.                       case 1: { int y = 7 * i; break; }
13.                       case 2: { Infinity inf = new Beyond();
14.                       Beyond b = (Beyond)inf; }
15.               }
16.  }
17. }
```

And given that line 7 will assign the value 0, 1, or 2 to sw, which are true? (Choose all that apply.)

A. Compilation fails

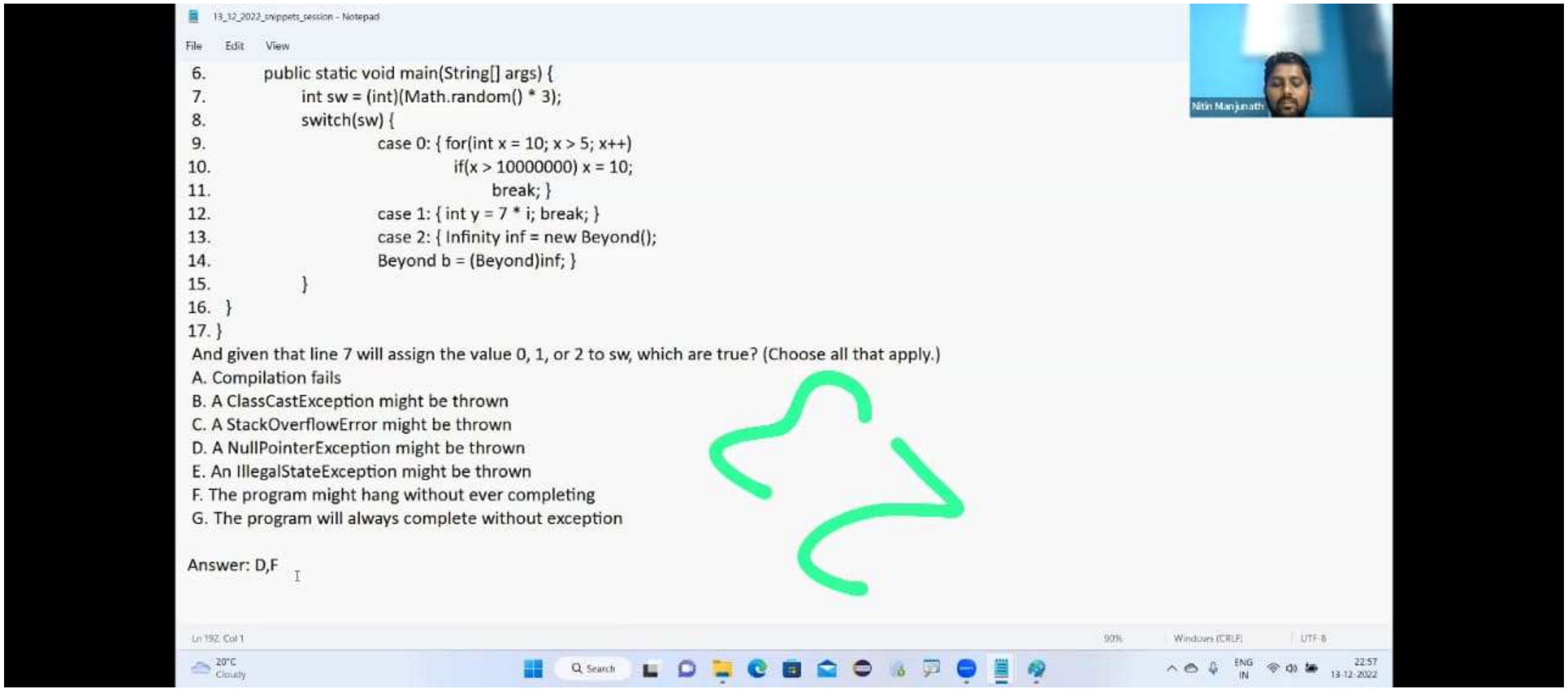
B. A ClassCastException might be thrown

C. A StackOverflowError might be thrown

D. A NullPointerException might be thrown

E. An IllegalStateException might be thrown

22



```
6.        public static void main(String[] args) {
7.            int sw = (int)(Math.random() * 3);
8.            switch(sw) {
9.                    case 0: { for(int x = 10; x > 5; x++)
10.                           if(x > 10000000) x = 10;
11.                               break; }
12.                   case 1: { int y = 7 * i; break; }
13.                   case 2: { Infinity inf = new Beyond();
14.                   Beyond b = (Beyond)inf; }
15.           }
16.   }
17. }
```

And given that line 7 will assign the value 0, 1, or 2 to sw, which are true? (Choose all that apply.)

A. Compilation fails
B. A ClassCastException might be thrown
C. A StackOverflowError might be thrown
D. A NullPointerException might be thrown
E. An IllegalStateException might be thrown
F. The program might hang without ever completing
G. The program will always complete without exception

Answer: D,F

23