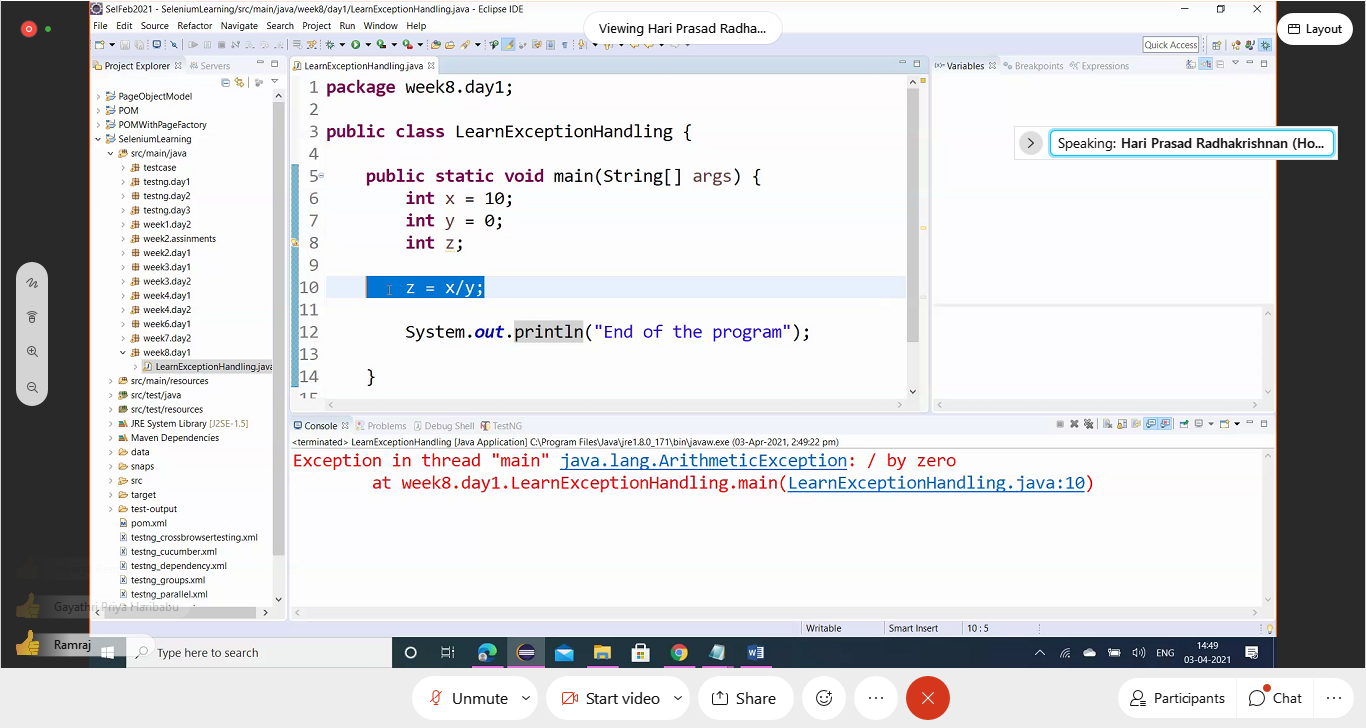
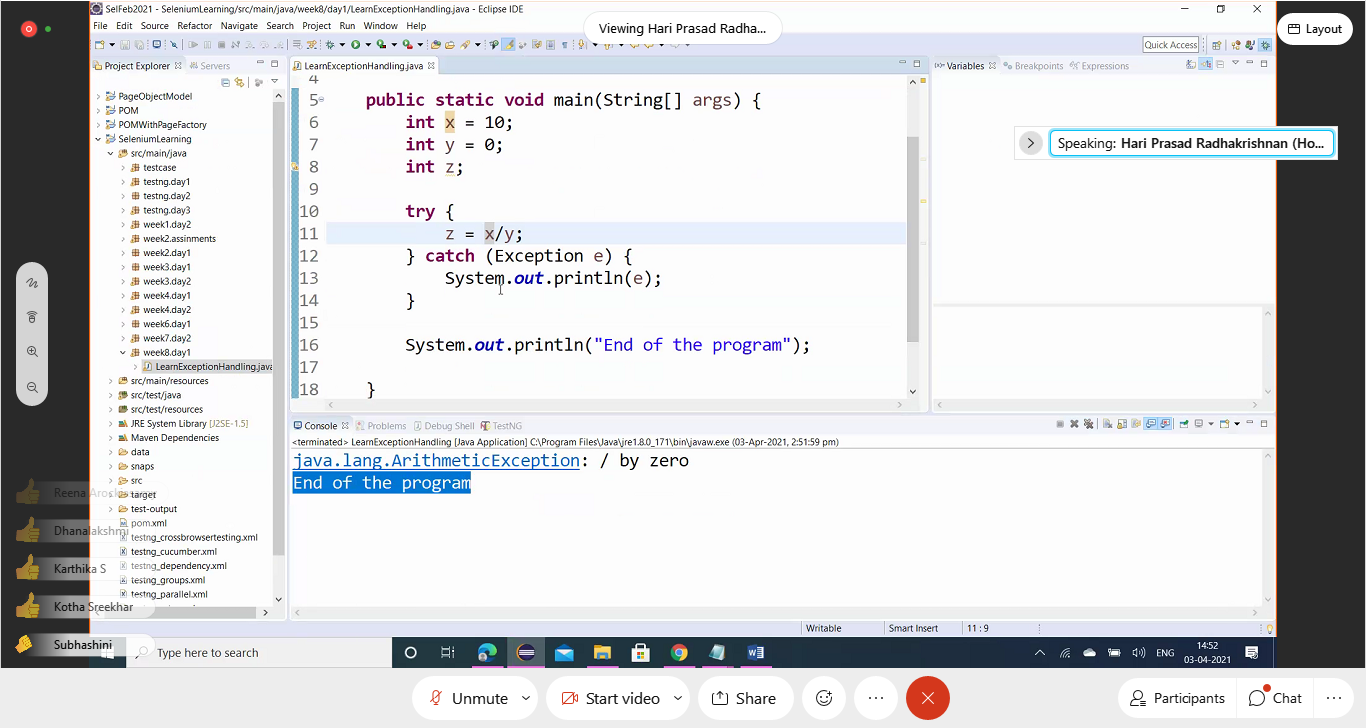
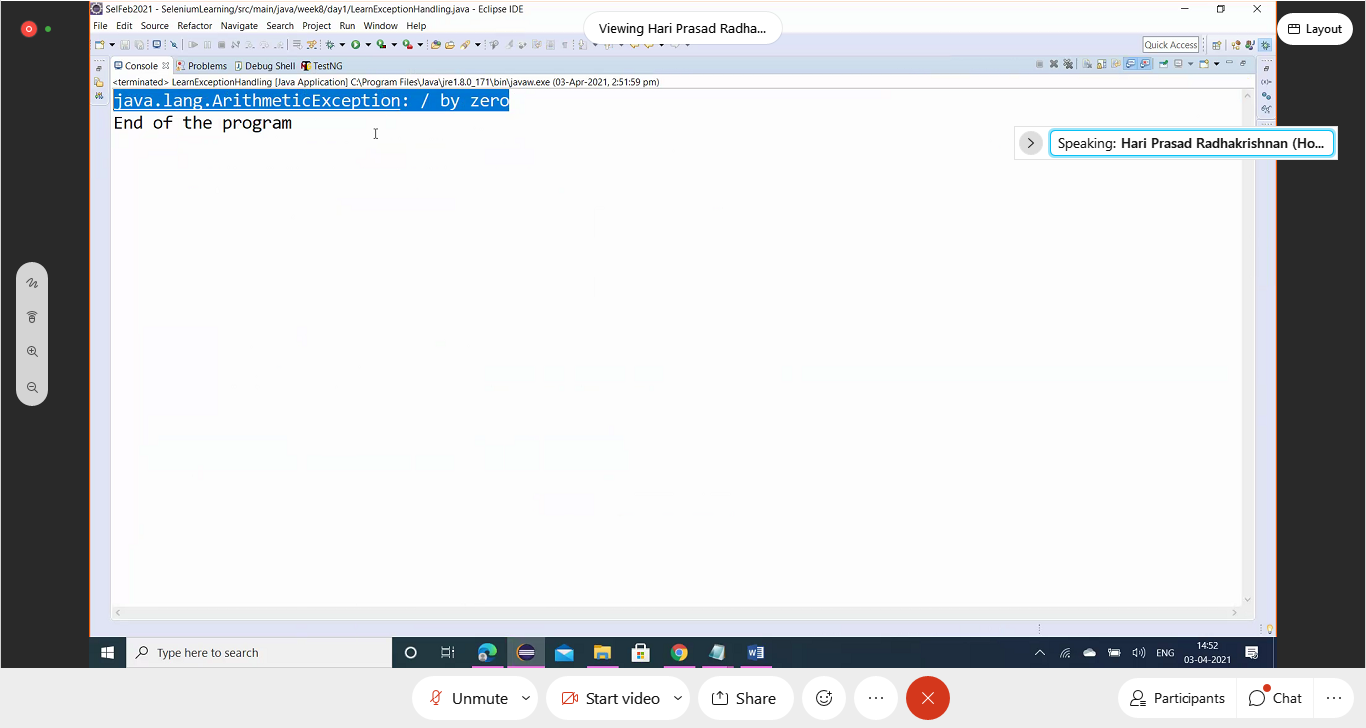


Exception thrown and the program is terminated abruptly.



Apply Try catch Block- It helps to continue the execution even after the exception. Catch block executes only when Exception is thrown:

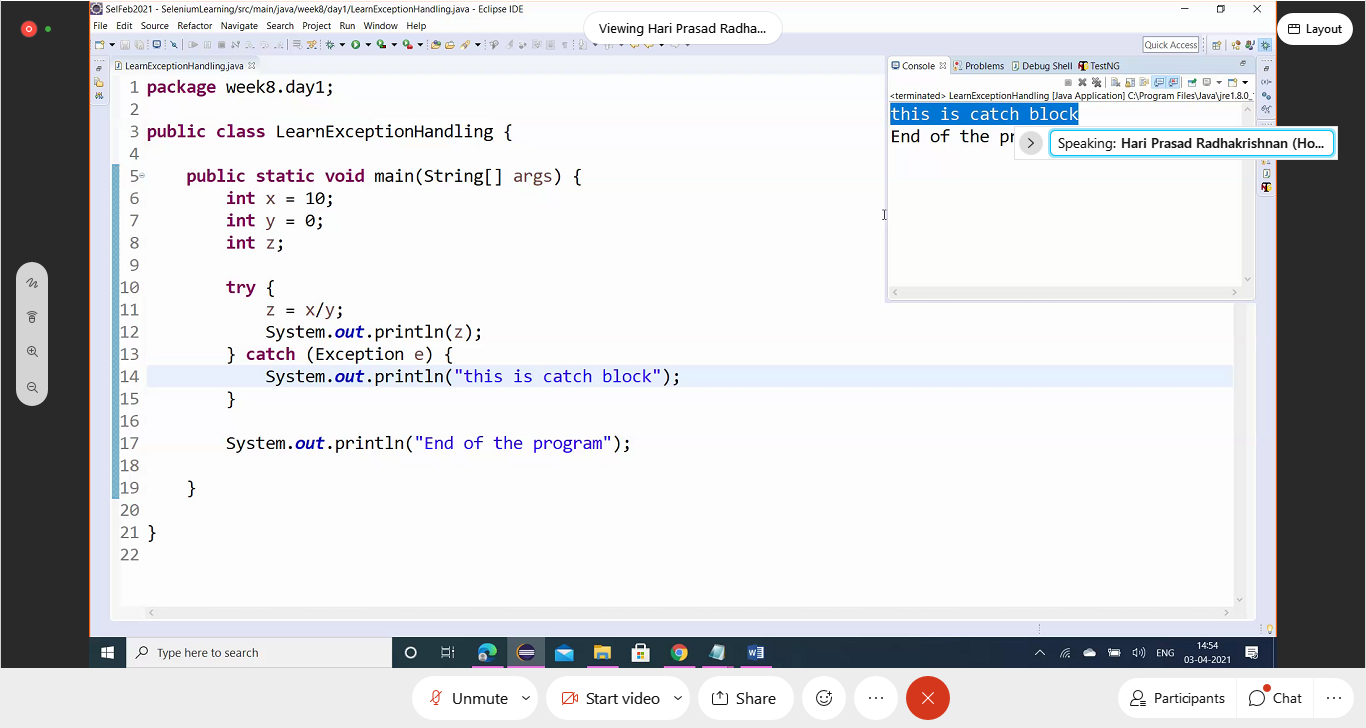




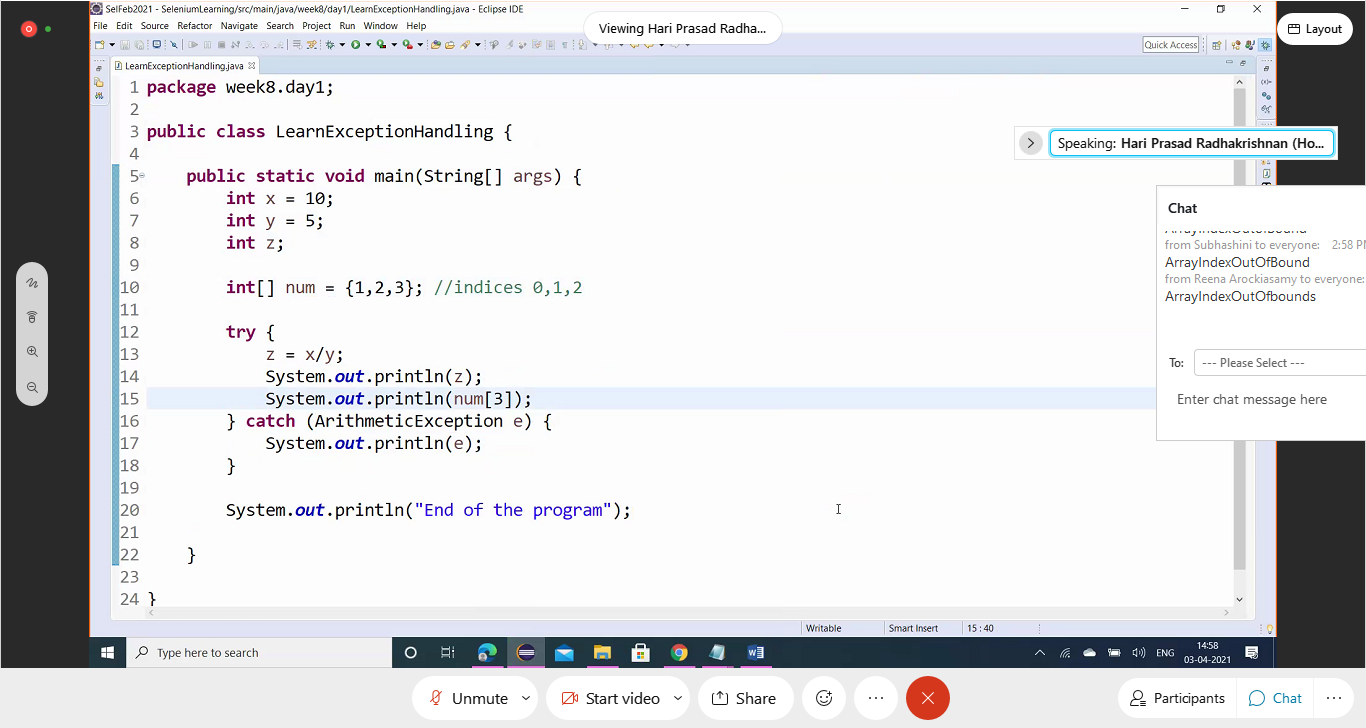
if there are any issues, then try block will get executed right? In this code , if y=0, then how do the sysout statement in tryblock behave?

The control goes to the catch block when an exception is encountered. The very next line that threw the exception will not be executed.

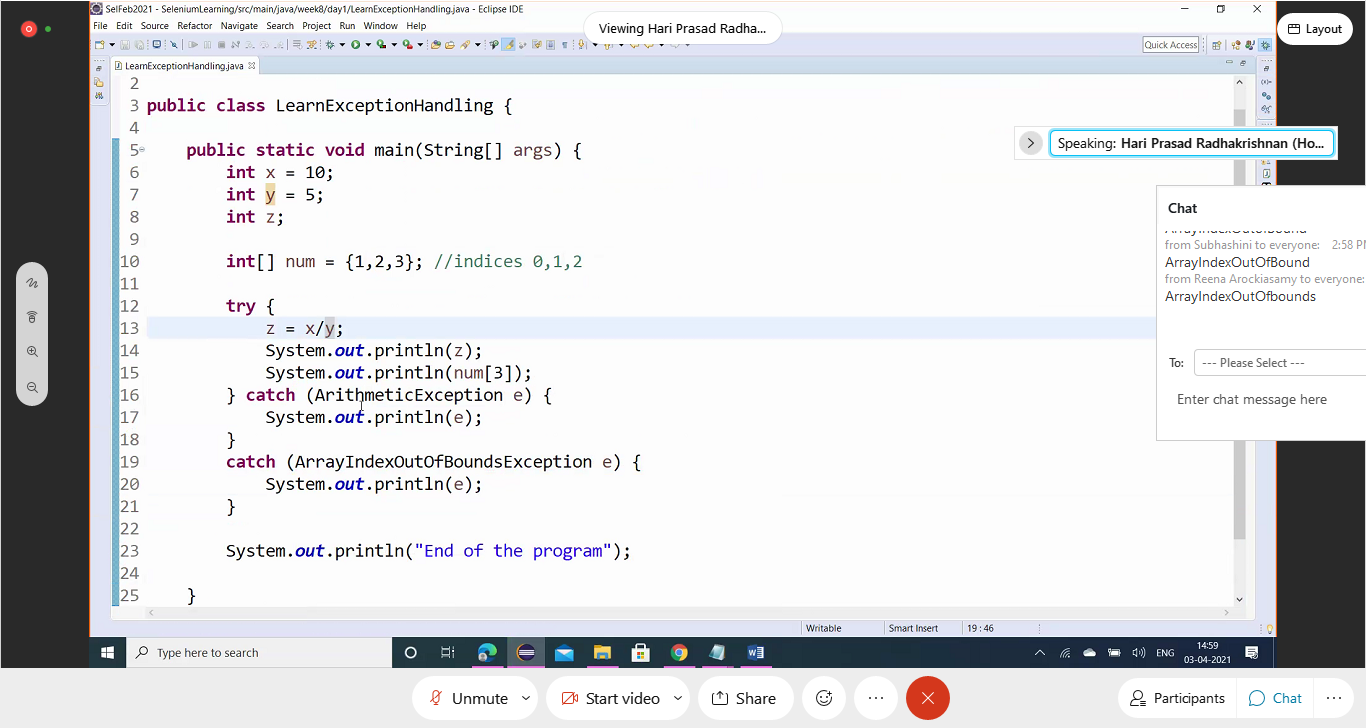
Catch block can have any part of code in it to be executed. Not just printing the exception details:



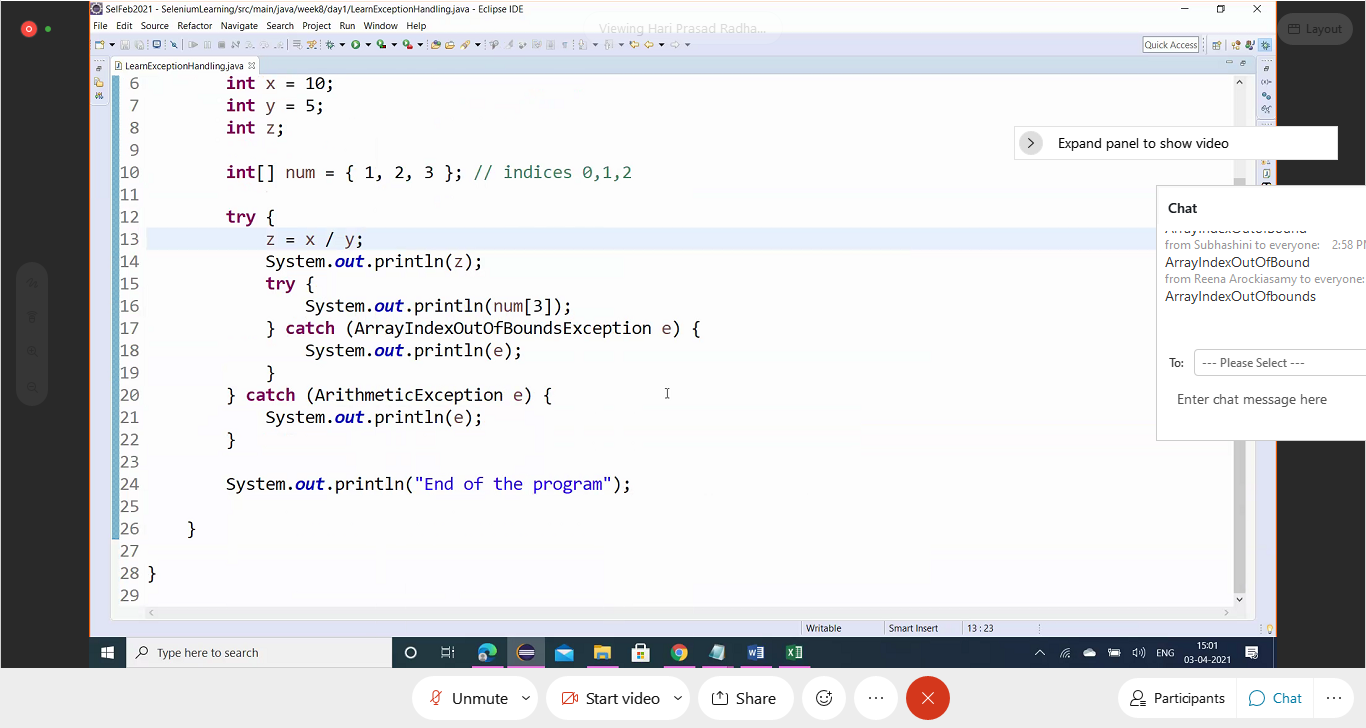
Specific Exceptions can be given in the catch block like Arithmetic exceptions ( TRY-CATCH)



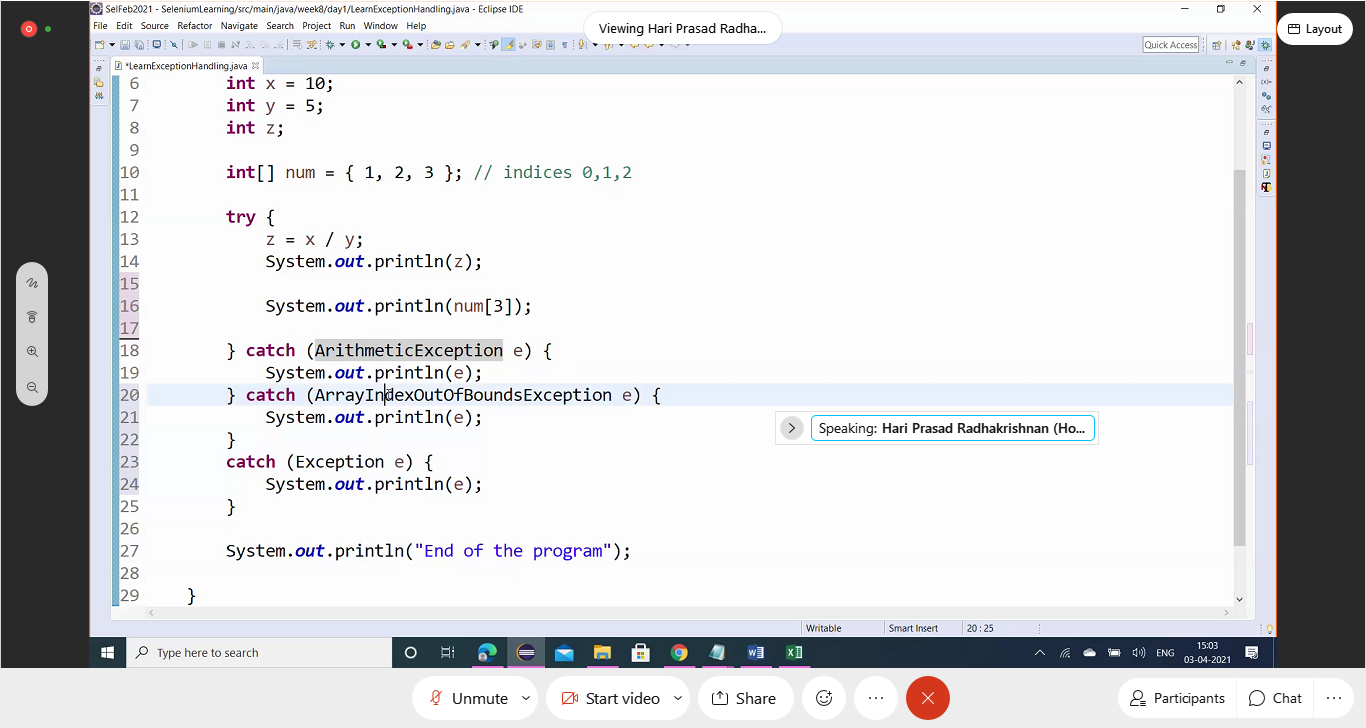
The above does not catch the ArrayIndexOutofBoud Exceotion. So, include multpole catches to handle the needed exception ( MULTIPLE CATCH BLOCK)



( NESTED TRY-CATCH)



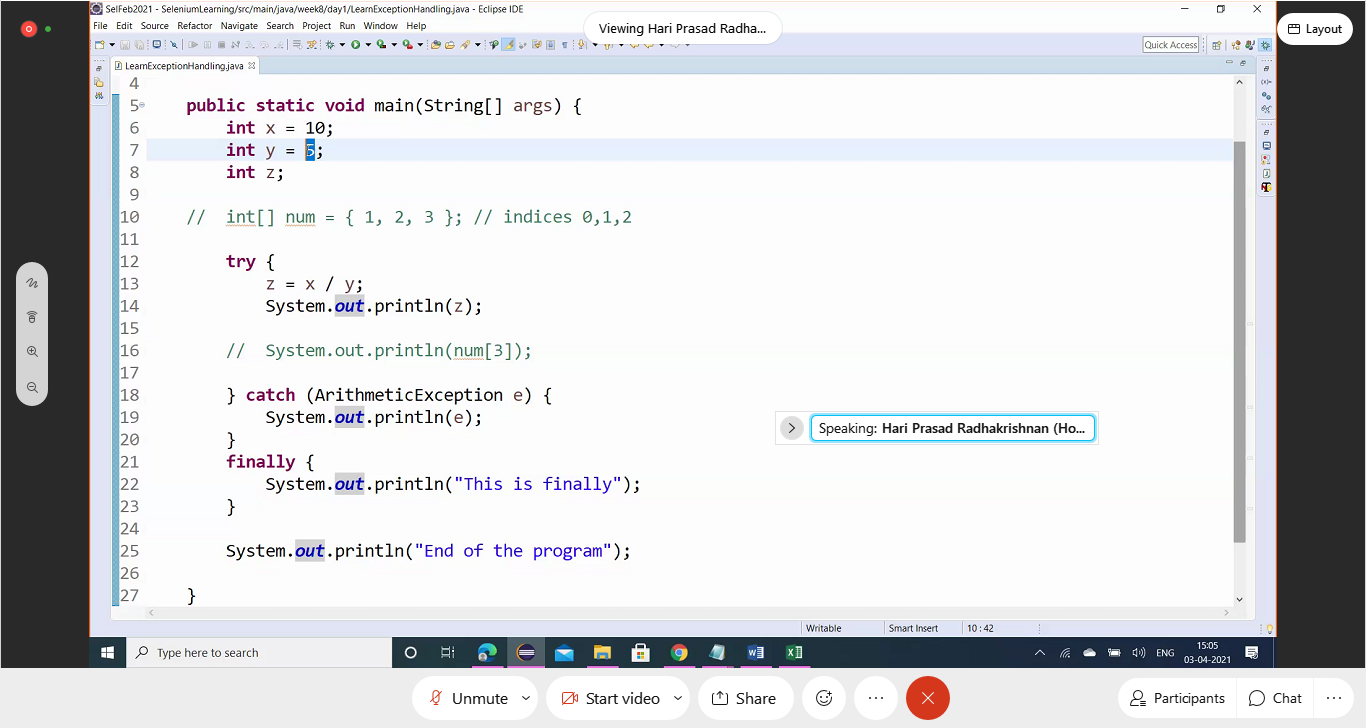
**Exception e** must be the last in the hierarchy of the multiple catch



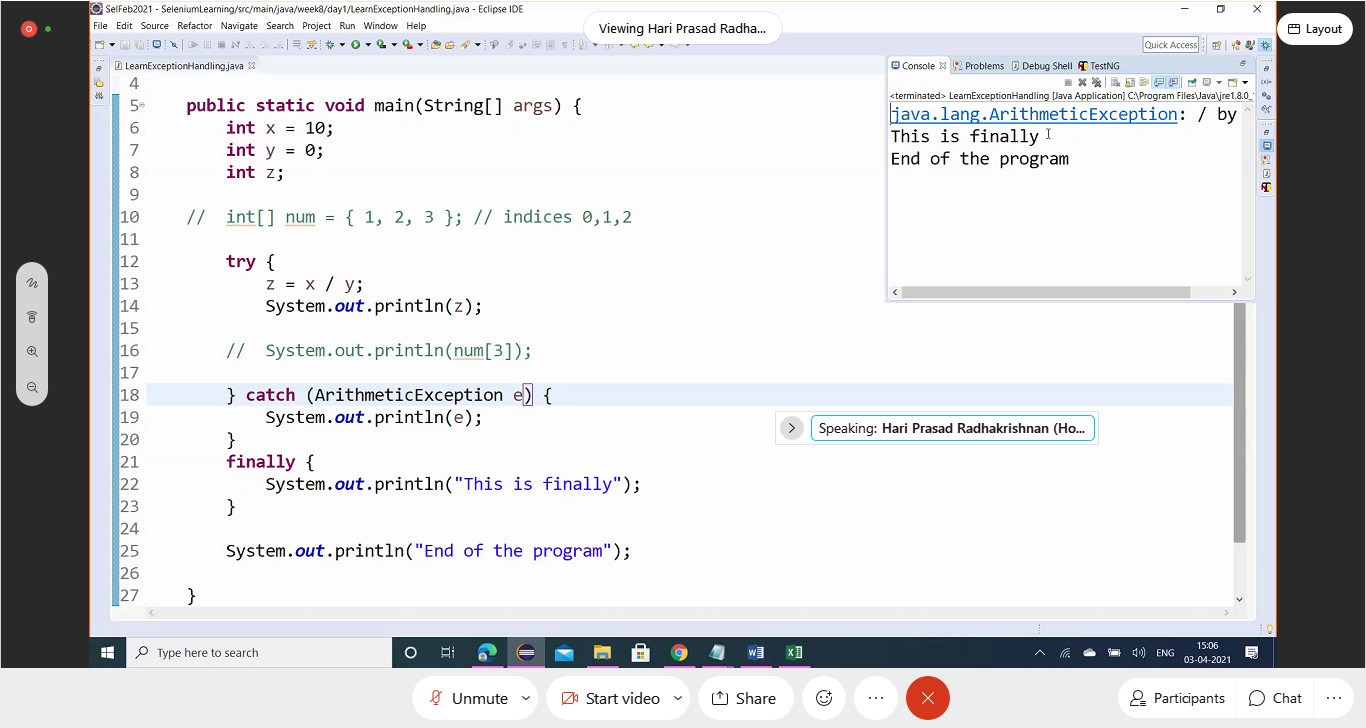
FINALLY BLOCK:

The code inside this block is always executed.

When there is no exception:

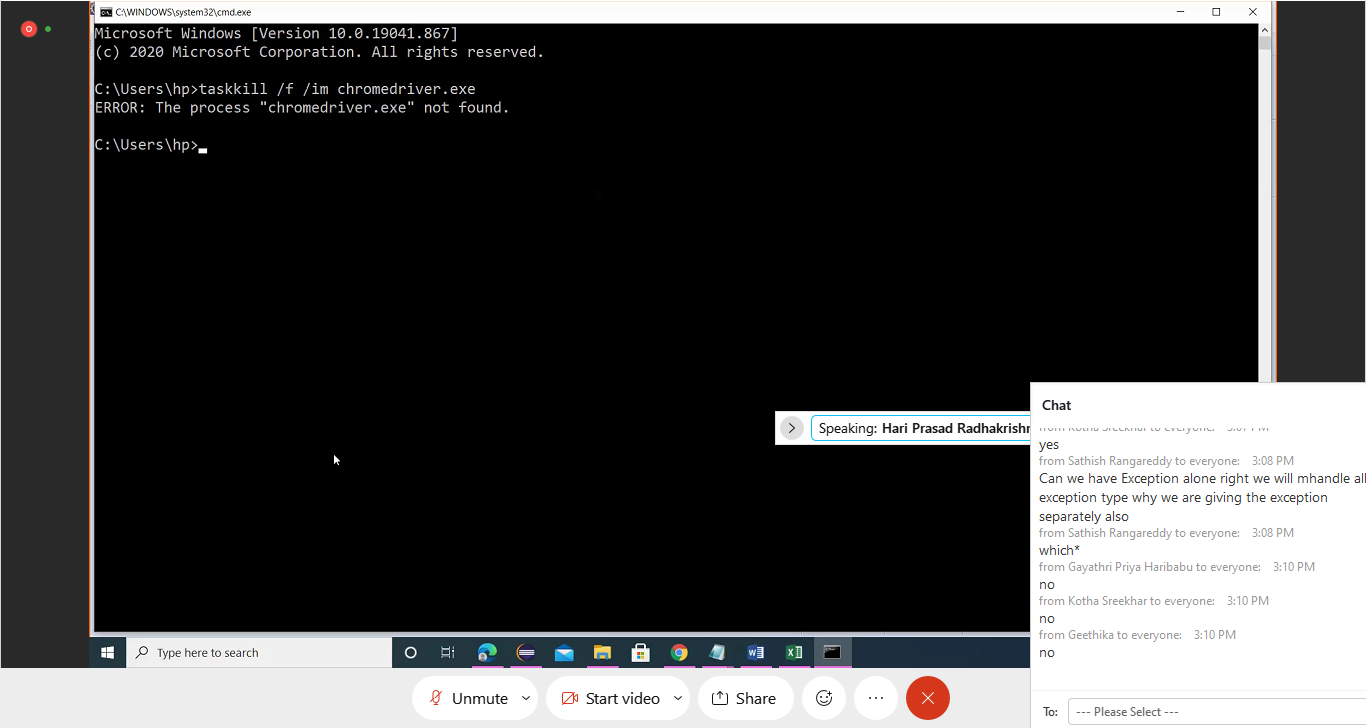


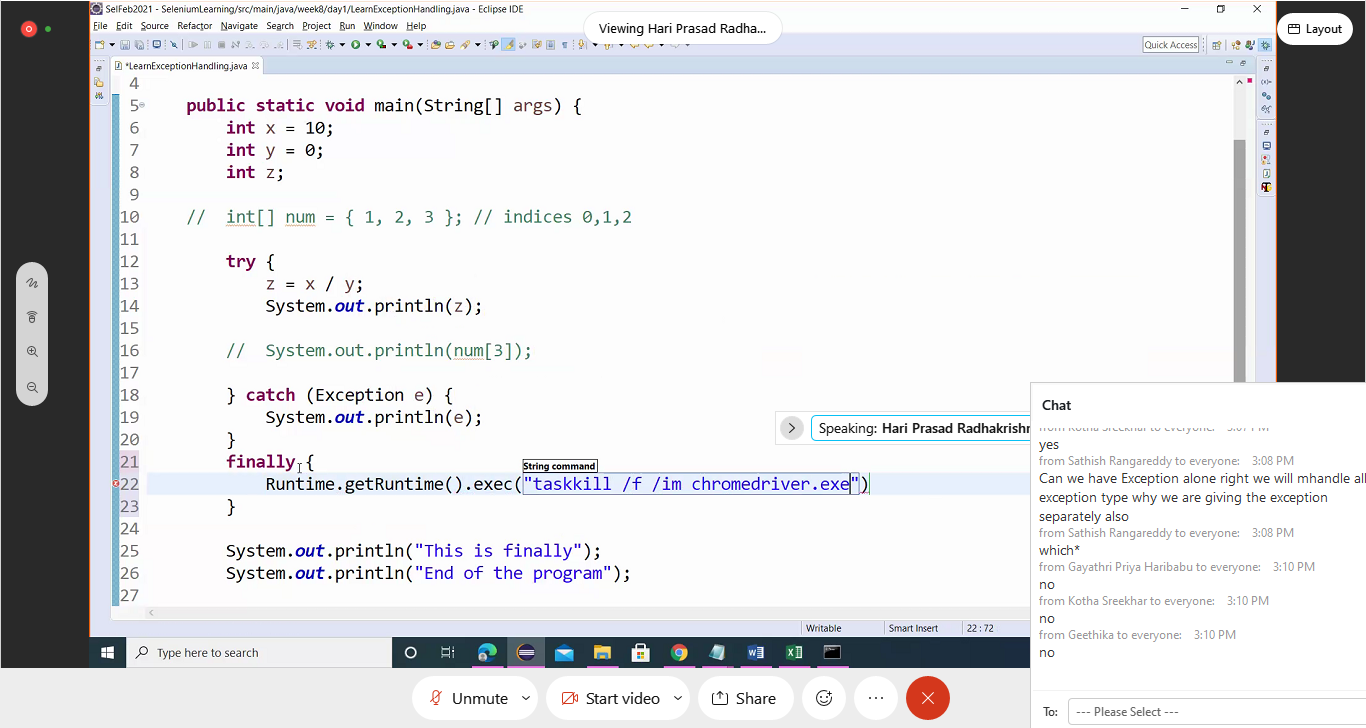
Even when the Exception occurs, Finally block is executed: ( We can also have Try and Finally alone without catch)



**Why we need finally, anyway next step going to run and we can give the code after catch block right?**

Kill the browser at any cost even if the exception the exception occurs. Anyway this will handled in the AfterSuite.



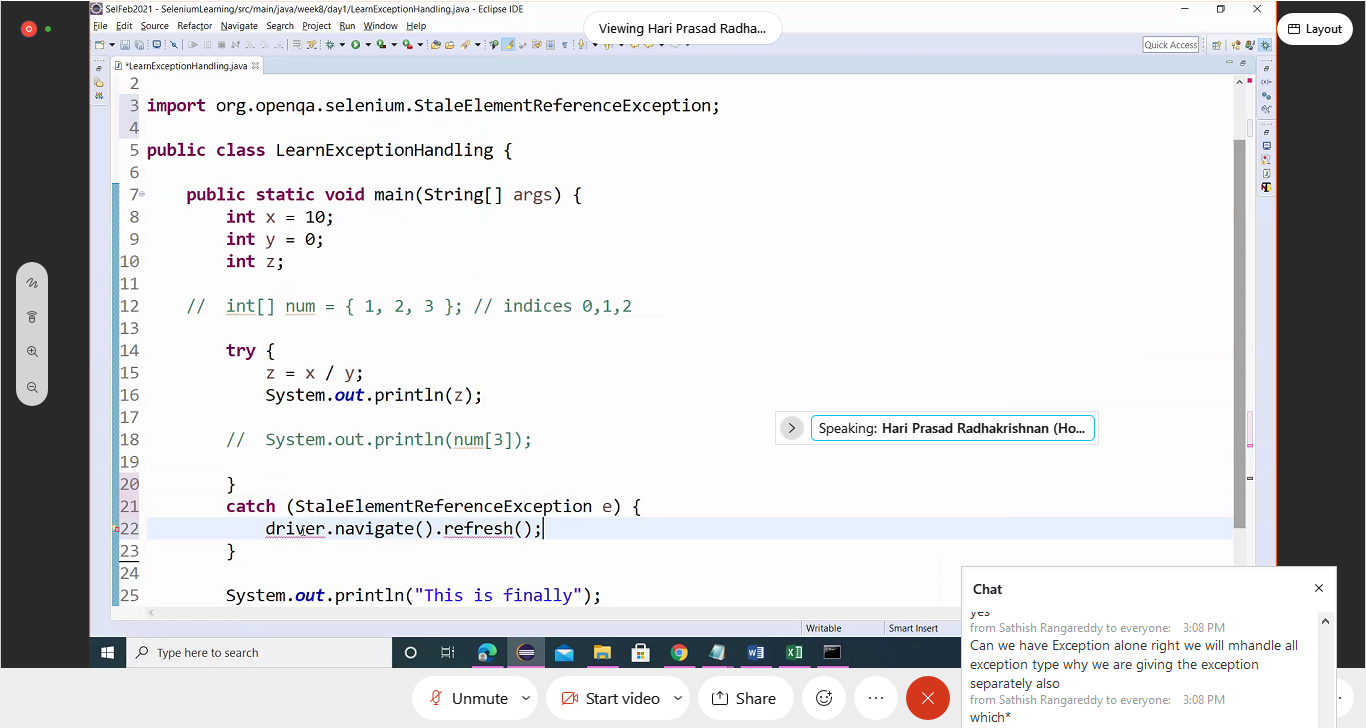


**Can we have Exception alone which handle all exception type? Why we are giving the exception separately also?**

Say, we get a StaleElementReferenceException. ( can be handled by increasing the wait or refresh).

In such cases we can give specific handling codes inside it .

By refreshing driver.navigate.refresh() or Thread.sleep

****

