# Java

## Exception

Define — error, mistake

Intro — handling, manipulation

compiler ~~error~~ Exception

new Exception

Syntax: Small try catch

multiple catch

Nested try-catch

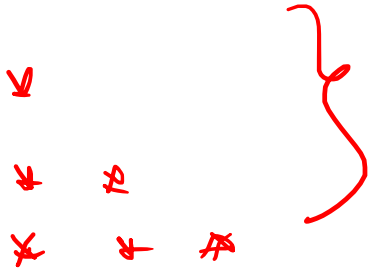
throw, throws does

Finally ?

user defined exception.

# "Managing errors and exception"

Human - mistakes comⁿ during pgm.



skeleton → run → errors → Debug.

qᵗʸ

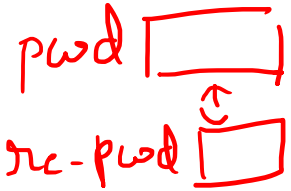
add

pgm halts execution

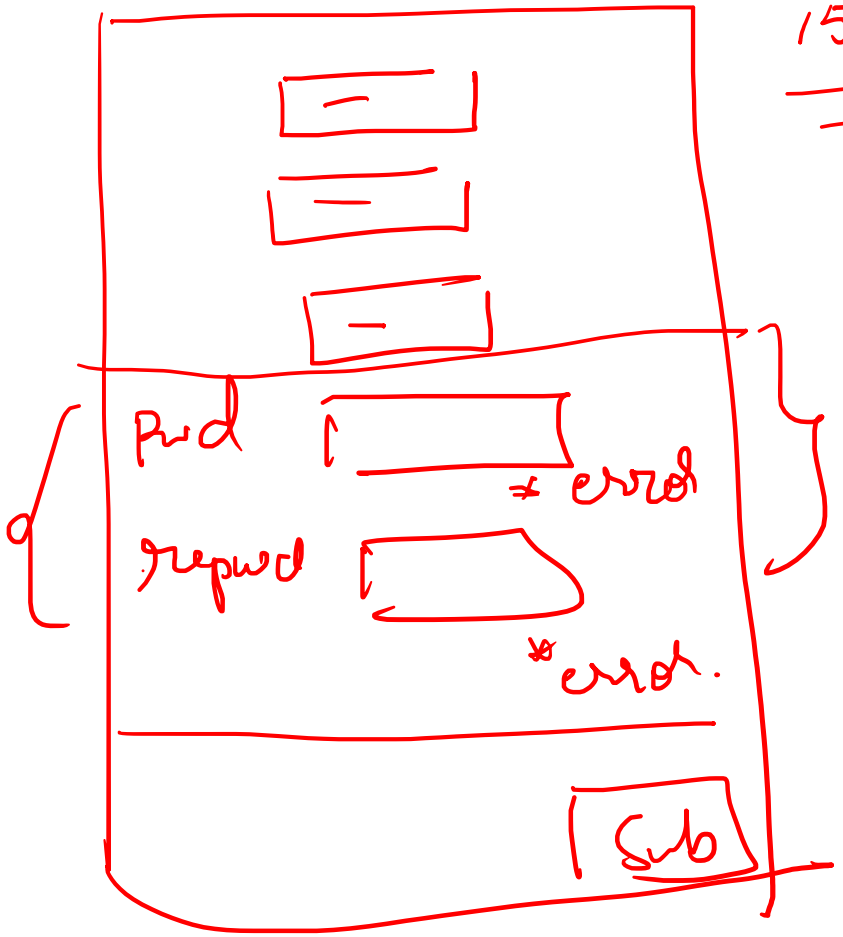
Banking Applicⁿ

Registerⁿ page

15-20 field




Terminates.




15 - 20 field

→ reset

done

Banking Appl.

↓

a Registration page

Begin

↓

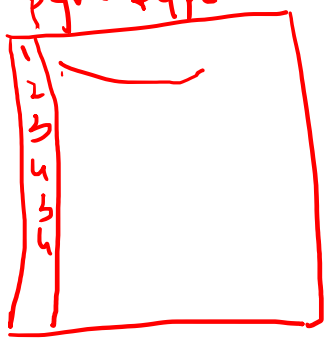
not a user friendly

↓ display → remain same → pausing

A mistake lead to error causes a pgm to produce unexpected results

↓

error — can stop execu| terminate




⚠ ⊗

Compiler error

↓

Syntax errors

No class found.

; ( ) { }

console

runtime error →

↓

execute → No expect result

↓

error.

logical, memory

'errors' - are noty but uncaught exception

("j", % zero)

→ exception → Handle errors (not manipulation)

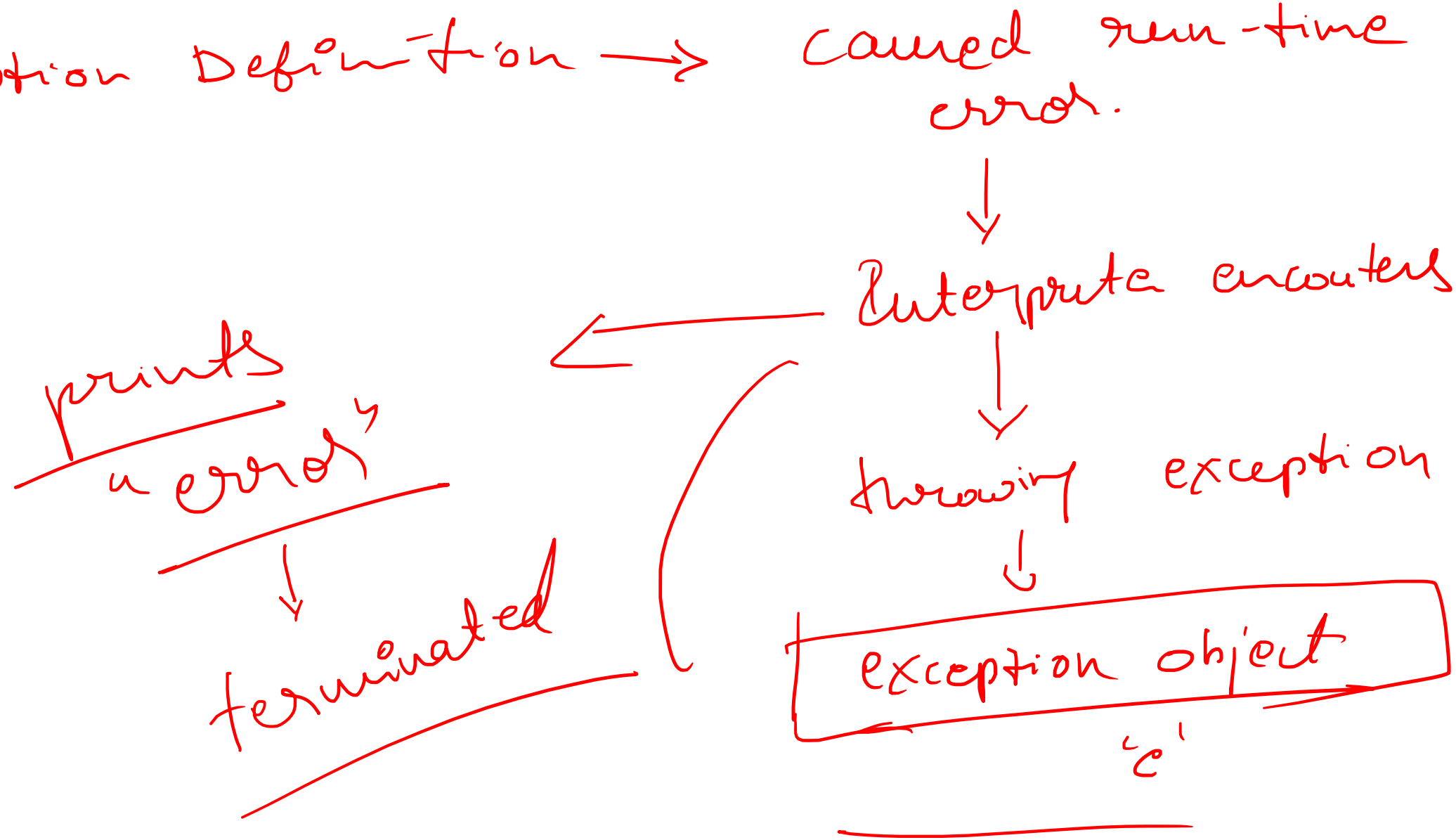
try : block of code that is sure error →

catch : handle error.

"-1" → exception

```
if(x < 0)
    S.o.pL("-ve valu")
```

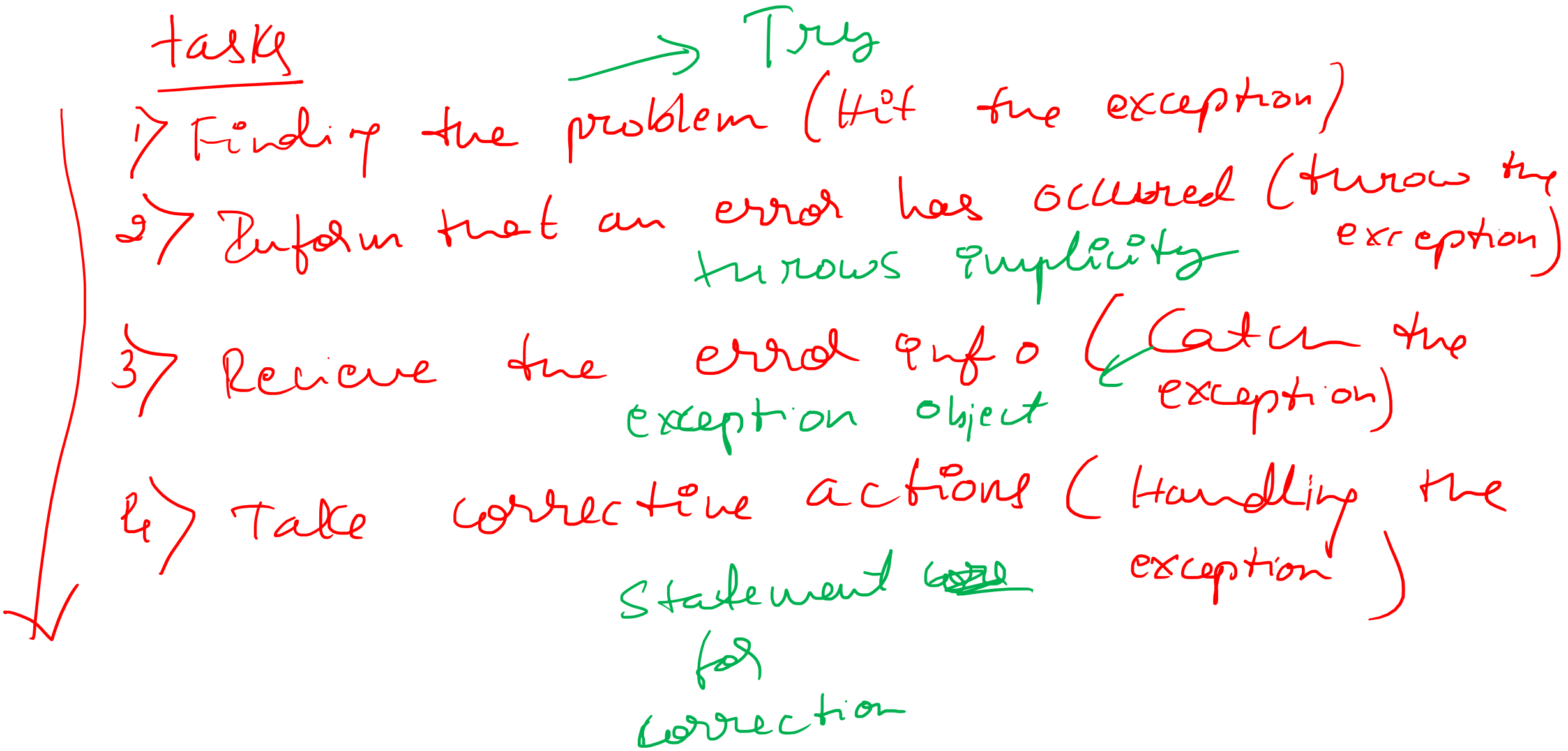
Exception Definition → caused run-time error.



# purpose of exception : ?

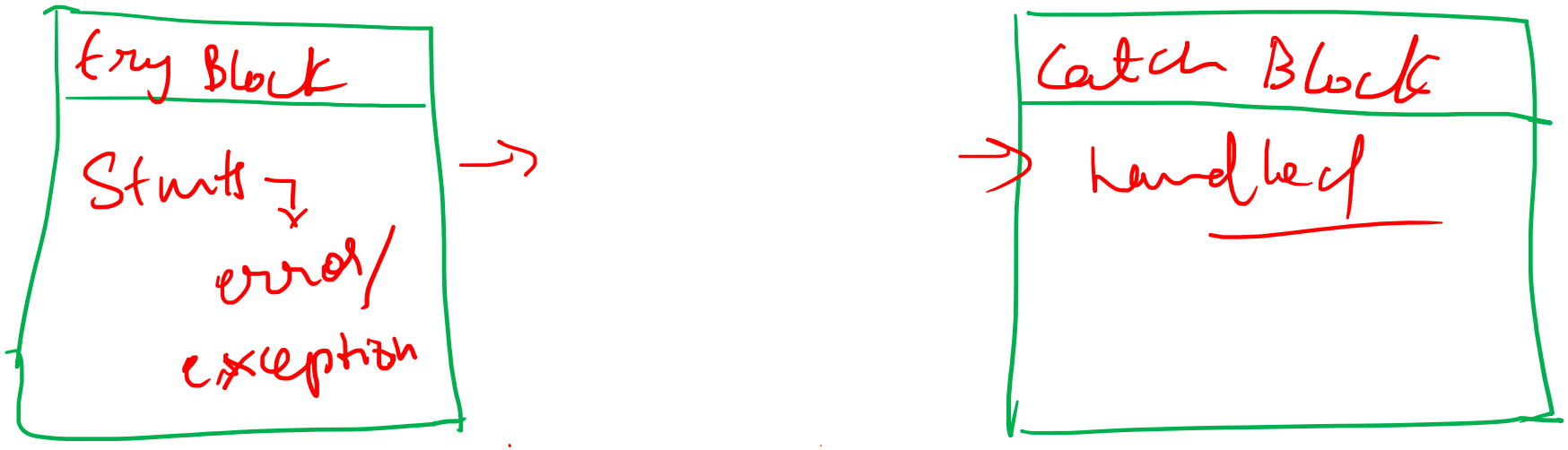
is to provide a means to detect and report an "exception circumstance" or error → Not terminate

handle


tasks
Try
1) Finding the problem (Hit the exception)
2) Inform that an error has occured (throw the exception)
throws implicitly
3) Recieve the error info (Catch the exception)
exception object
4) Take corrective actions (Handling the exception)
statement
for
correction

# Some of Java Comm^n Exception :

## Syntax for EH code. [Block dia]



# Syntax

```
try {
    % array {
    .....
}
catch(Exception_type e)
{
    .....
}
.....
```

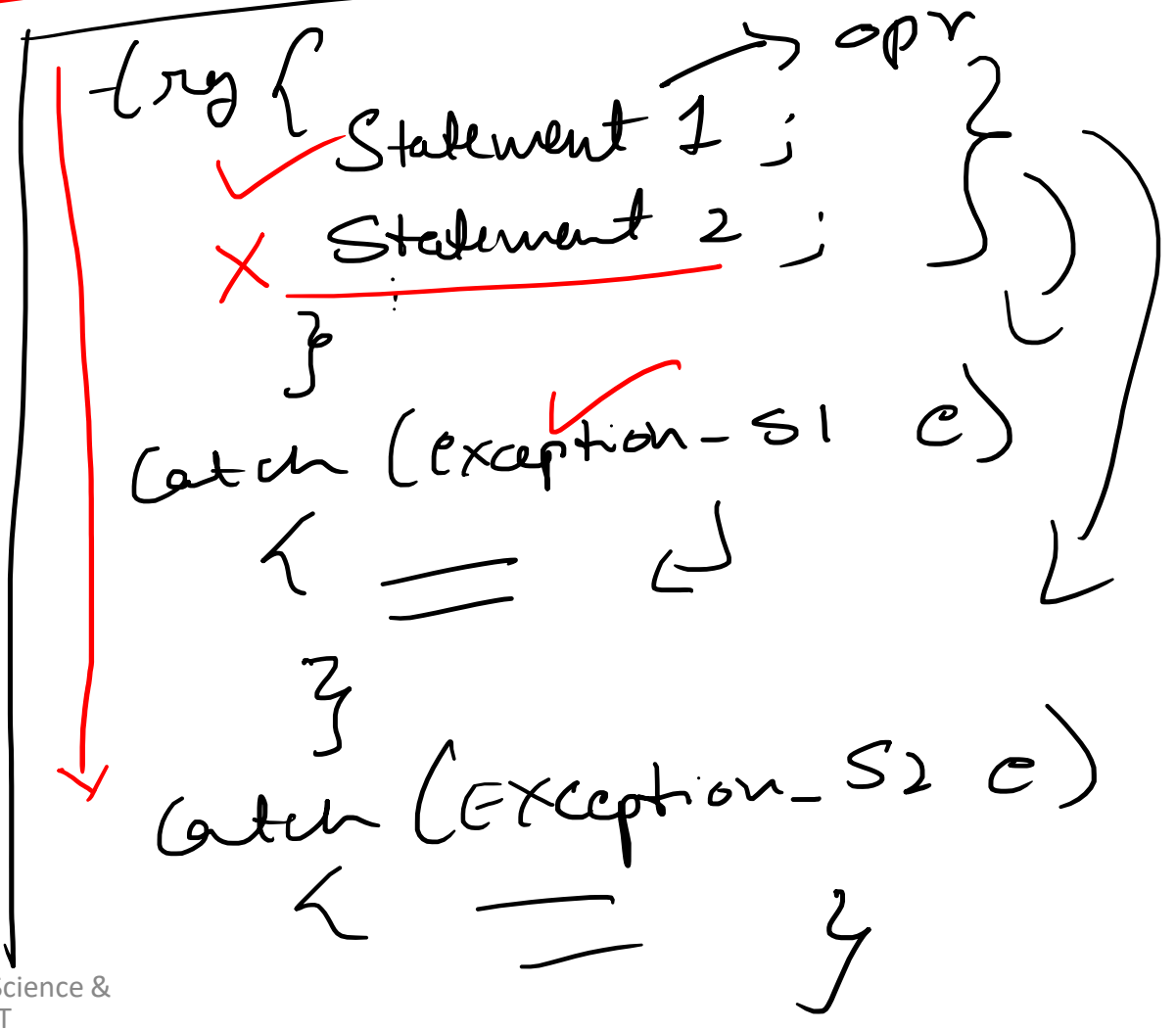
A block of code that is sure to cause an error + throws the error (Exception)

catches the exception thrown and handle it appriately

# Multiple catch Statement

```
try {
    a/0;
    int a = char;
}
catch(Exception e)
{
    ...
}
```

error → Exception



When an exception in try block is generated, the Java treats the multiple catch statement like cases in a switch statement

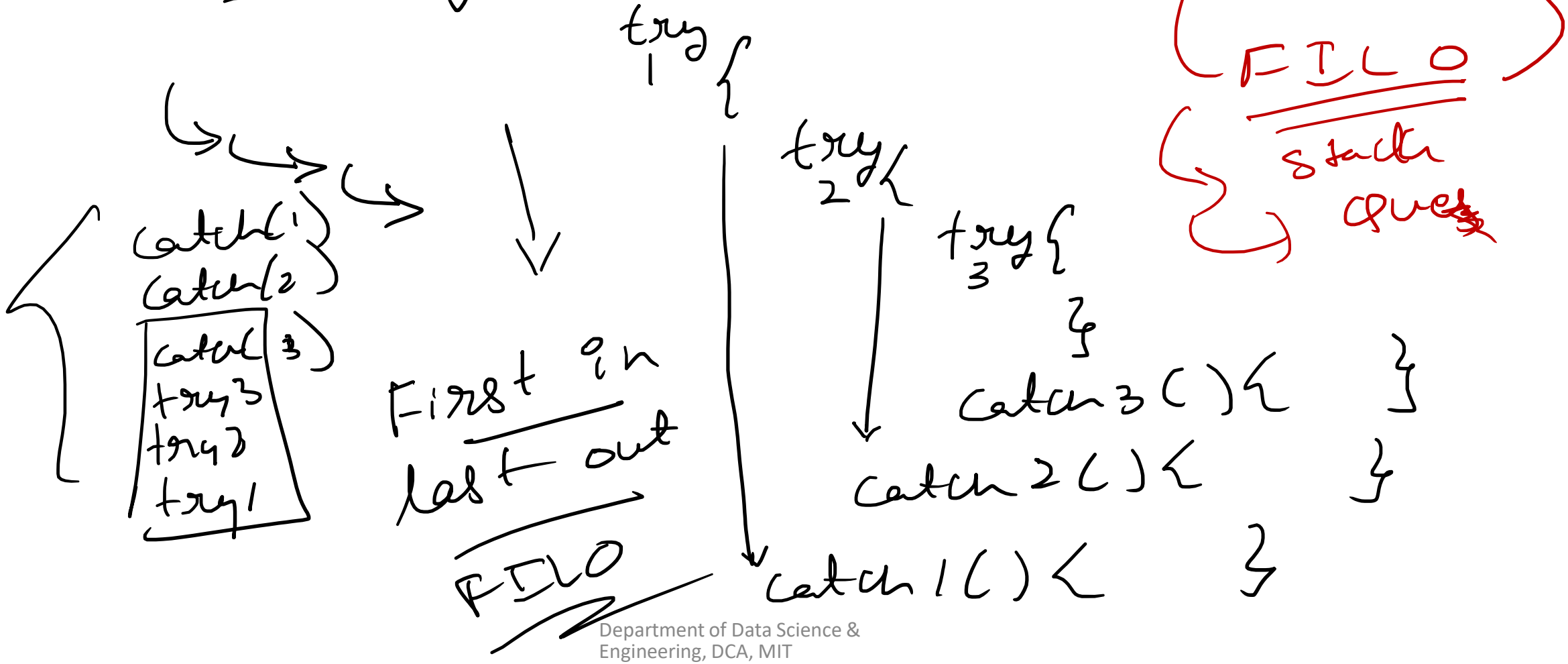
↓

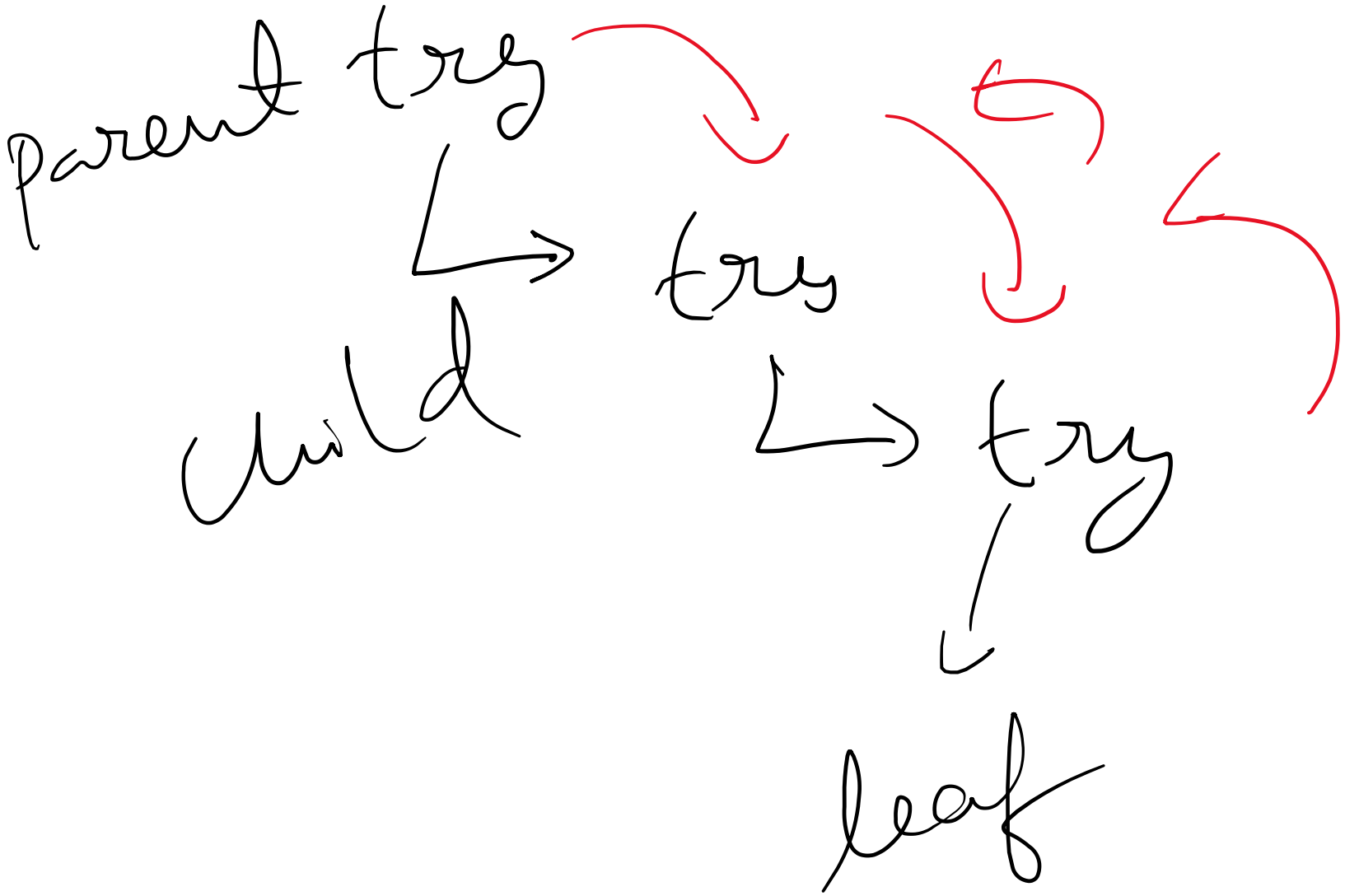
exception is caught.

The first stmt whose parameter matches with the exception object will be executed. & rest will be skipped.

# Multiple try - catch Statements

## Nested try-catch Block.




parent try
child
try
try
leaf

# Throw and throws → method.

Java

unchecked

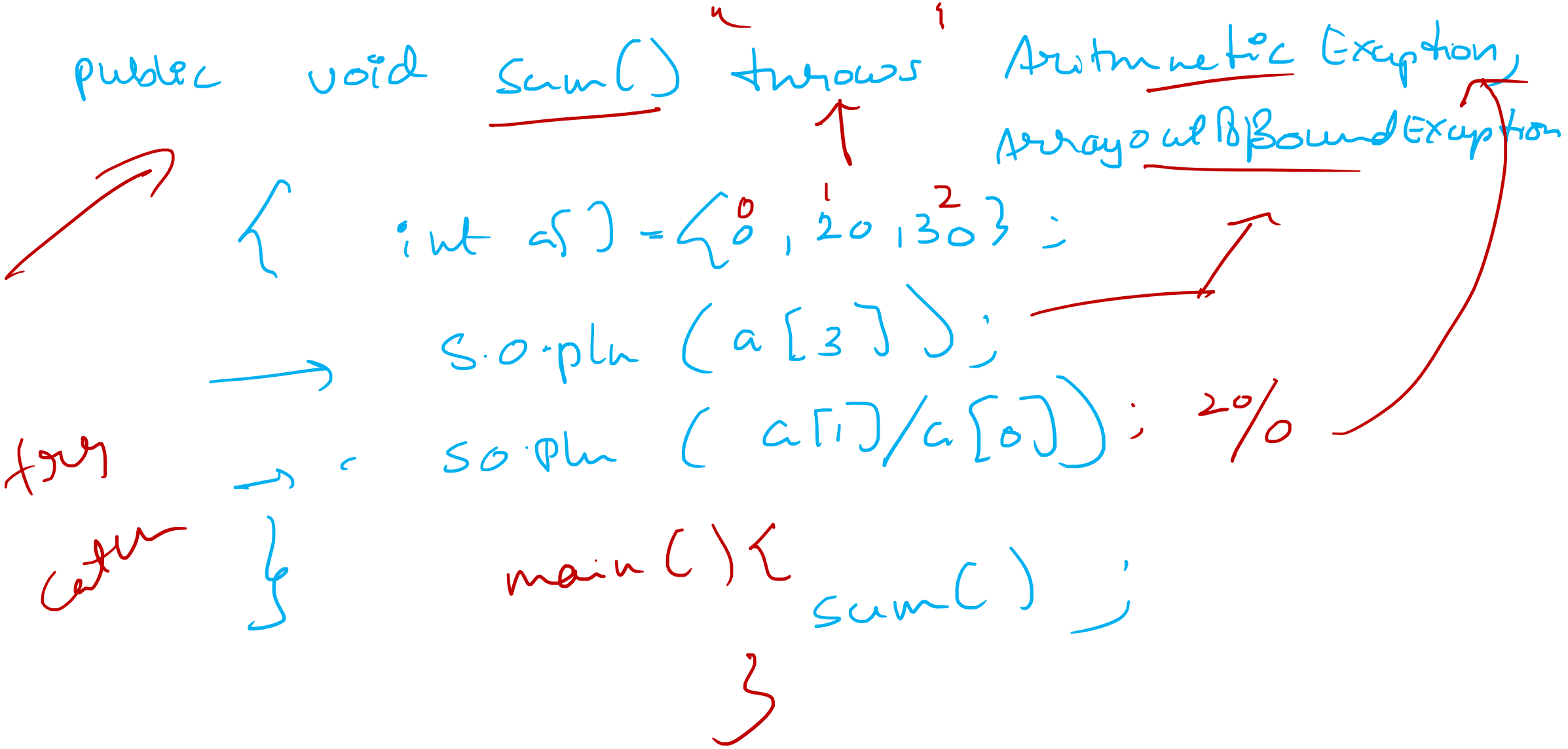
Arithmetic
% 0

checked

IOException - file i/p o/p
read()

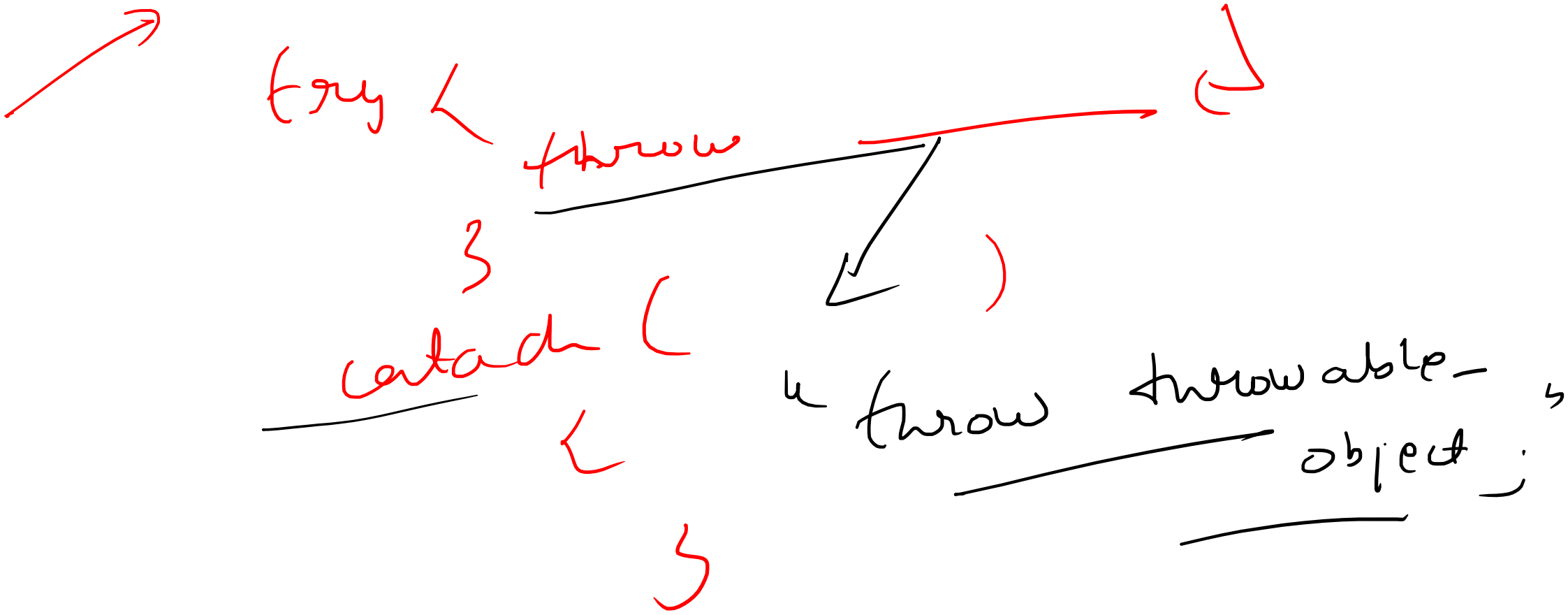
accessmodifier Datatype methodname() throws Ec1, Ec2

{ ≡ }


public void sum() throws ArithmeticException, ArrayOutOfBoundException
{ int a[] = {10, 20, 30};
S.O.Pln(a[3]);
S.O.Pln(a[1]/a[0]); 2%
try
catch
}
main(){
sum();
}

# Throw

→ Single exception - explicitly



## Finally :

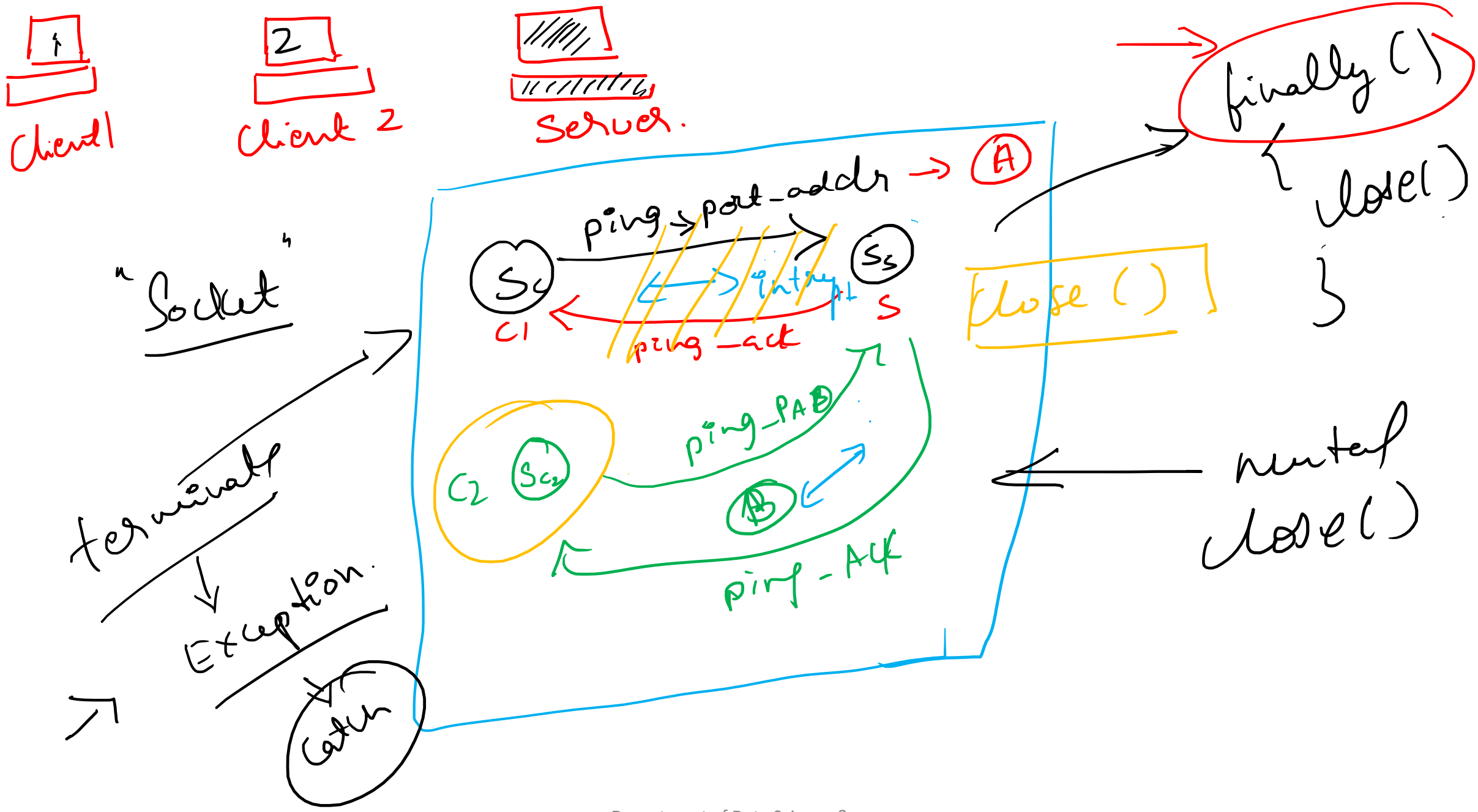

finally ( ) → clean up the code

→ connection termination

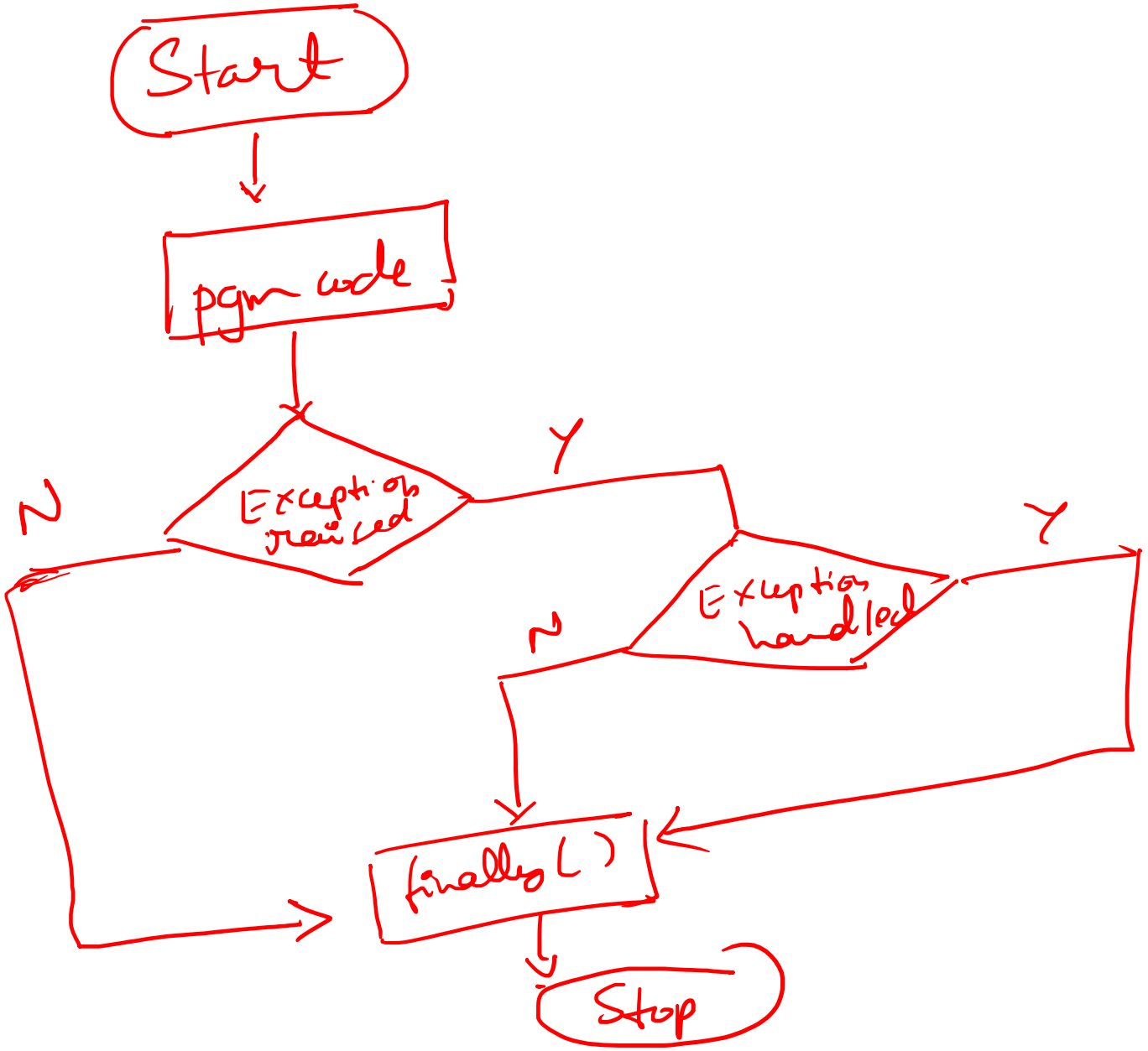
→ 'x' stmt to be executed event on try-catch terminate

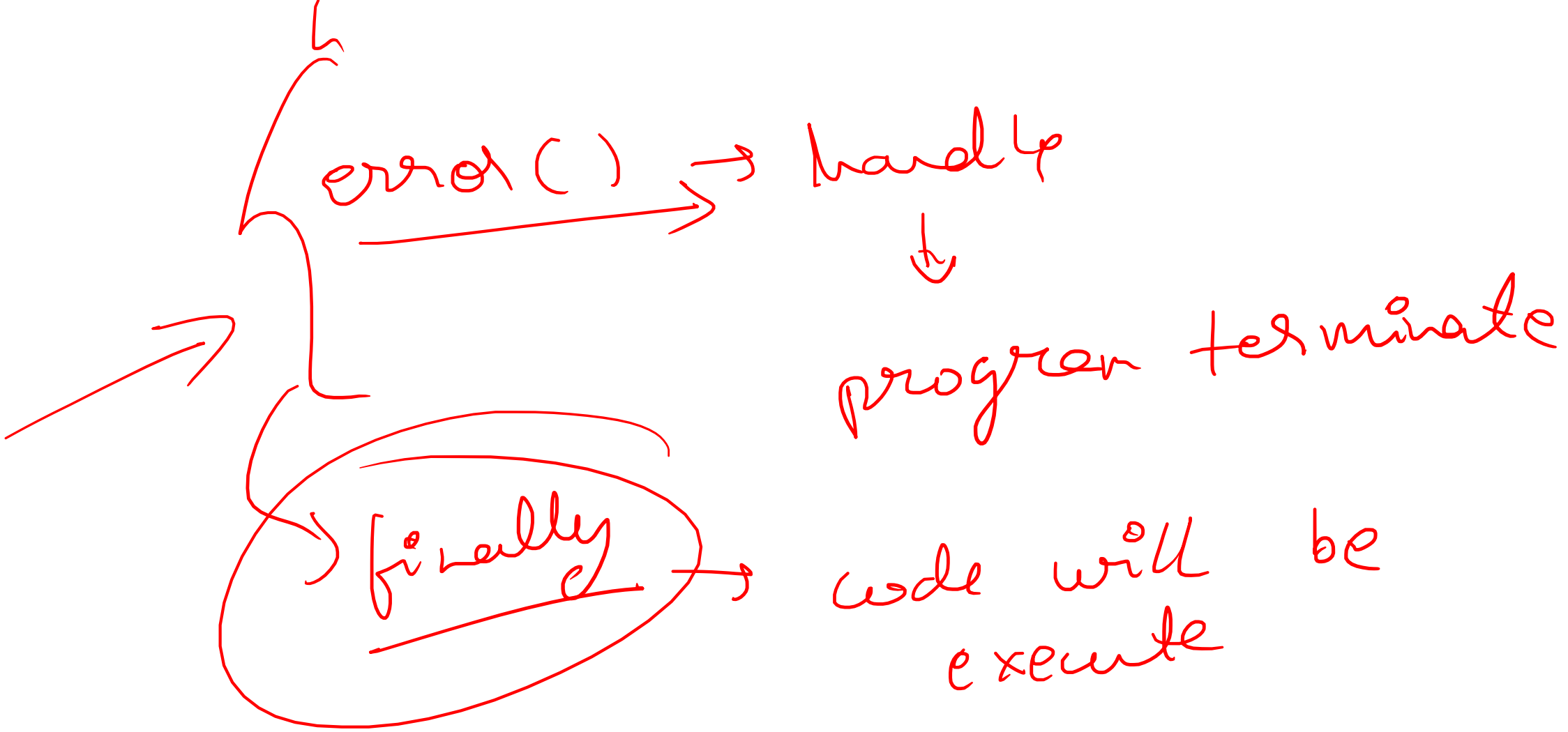
"closing" a connection

finally


1
2
Client1
Client 2
Server.
finally()
{ close()
}
"Socket"
ping → port_addr → A
ping
interrupt
ping_ack
C1
S
close()
terminate
Exception
catch
C2
ping_PAD
ping_ACK
numtcp
close()

Finally


Start
pgm code
Exception raised
Y
N
Exception handled
Y
N
finally ( )
Stop


error( ) → handle
program terminate
finally → code will be execute

# User defined Exception

Java → option → Exception

↓

a "throw"