

$\rightarrow$  classes whose objects doesn't exist  
(can't be created)

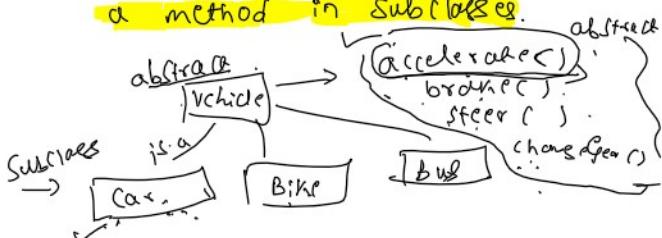
$\rightarrow$  abstract class Shape {

abstract area() :  $\downarrow$  Should be overridden.

{ one abstract method,

$\downarrow$  Impl. doesn't exist in base class  
 $\downarrow$  has to be overridden by sub class.

$\rightarrow$  enforce the implementation of a method in subclasses.



Interfaces: by default abstract

$\rightarrow$  all the methods are abstract

Public Interface Shape {  
void area() ; }



```

public interface Shape {
    void area();
    void display();
}

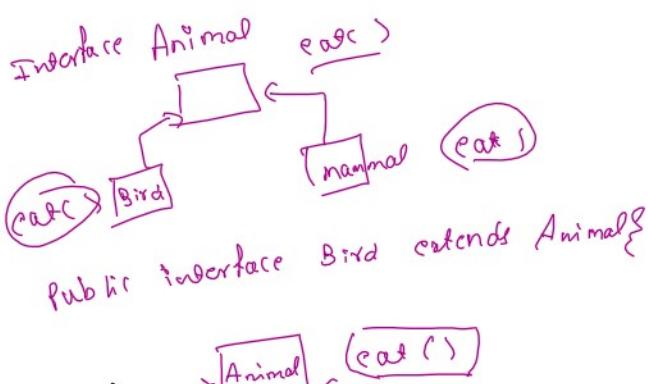
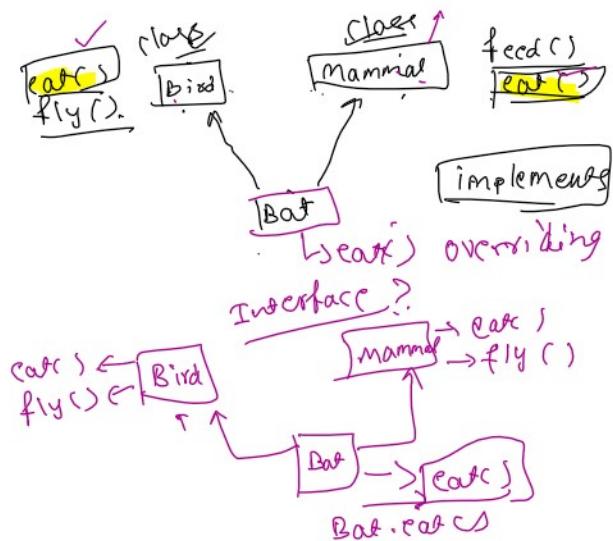
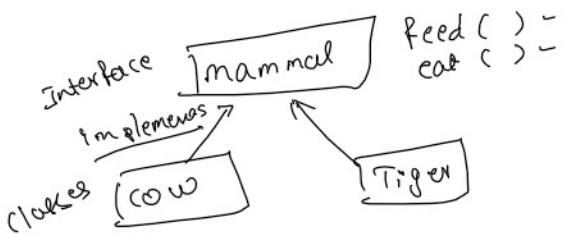
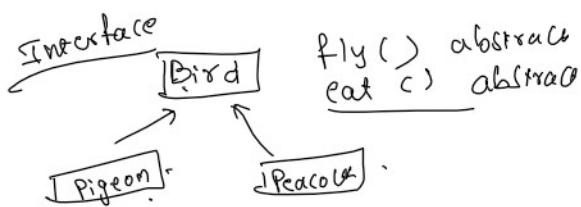
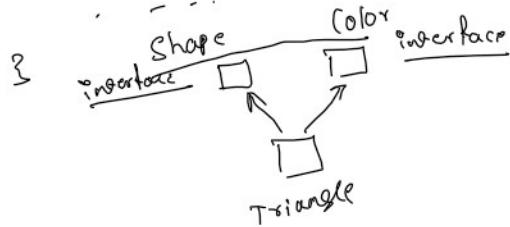
```

extends X  
implements

```

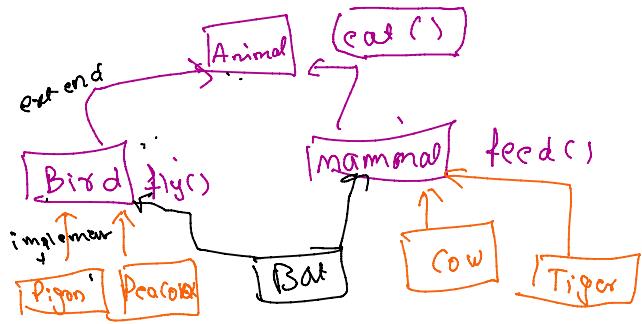
class Triangle implements Shape {

```





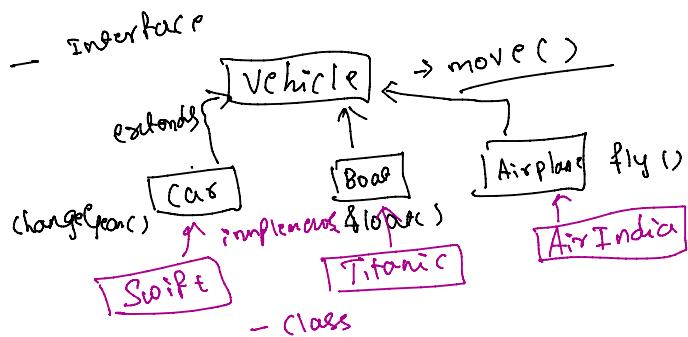
QUESTION



→ Class inheriting a Class  
↳ "extends"

→ Class implementing interface  
↳ "implements"

→ interface inheriting interface  
↳ "extends"

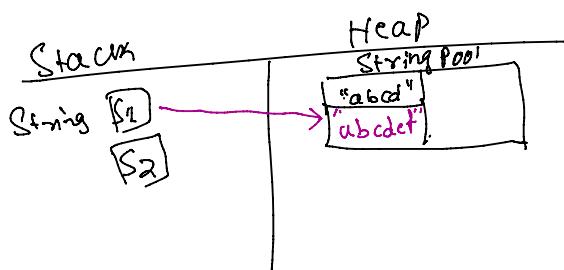


### String in Java

Class in java → data type

== "abcd" ---

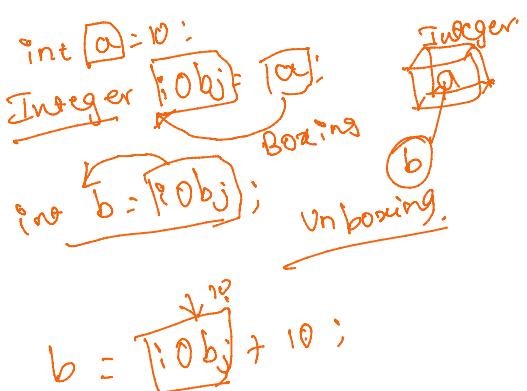
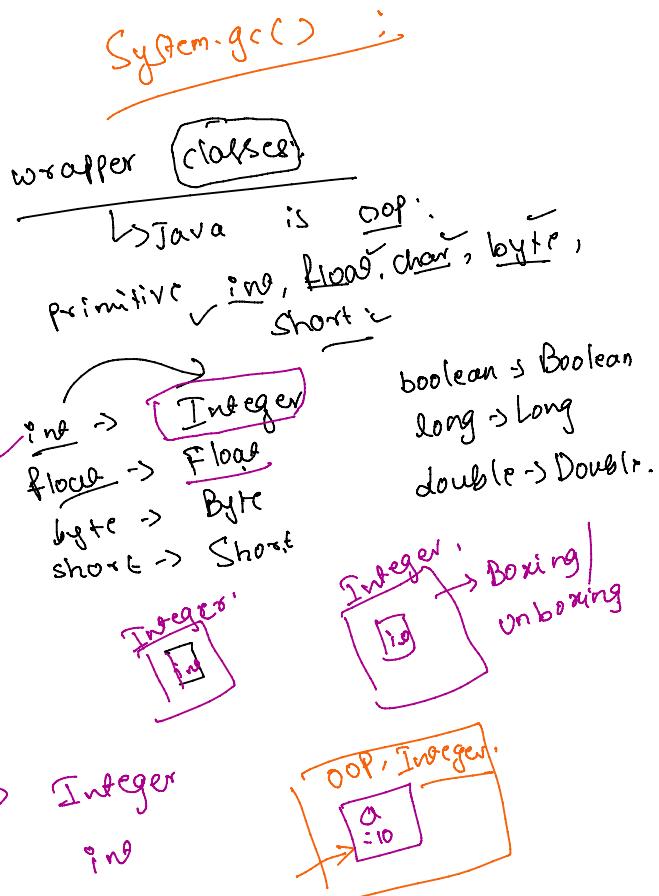
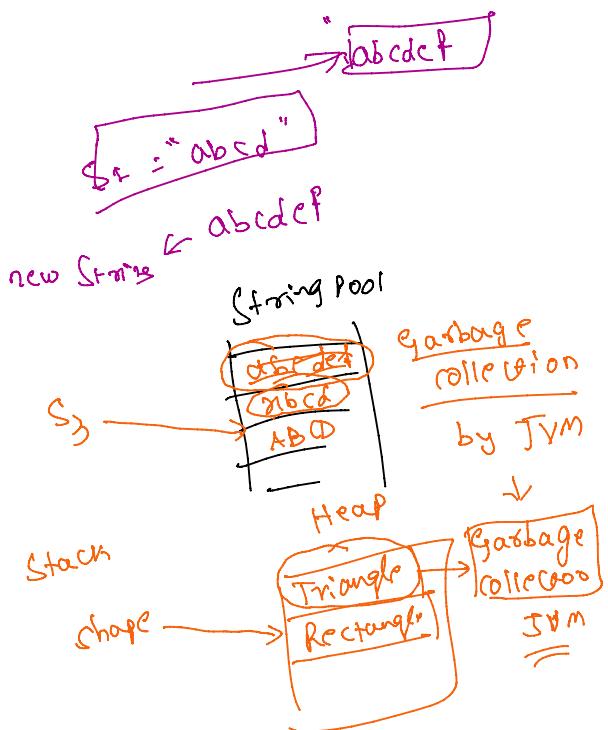
```
String s = "abcd";  
String s = new String(" ");  
= .toString();
```



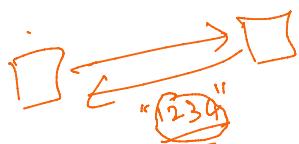
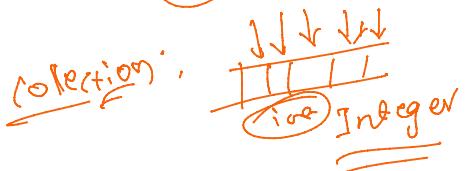
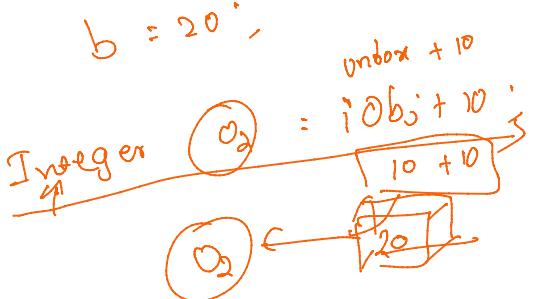
String is immutable

→ abcd ef









in a : Integer.parseInt("234")  
234

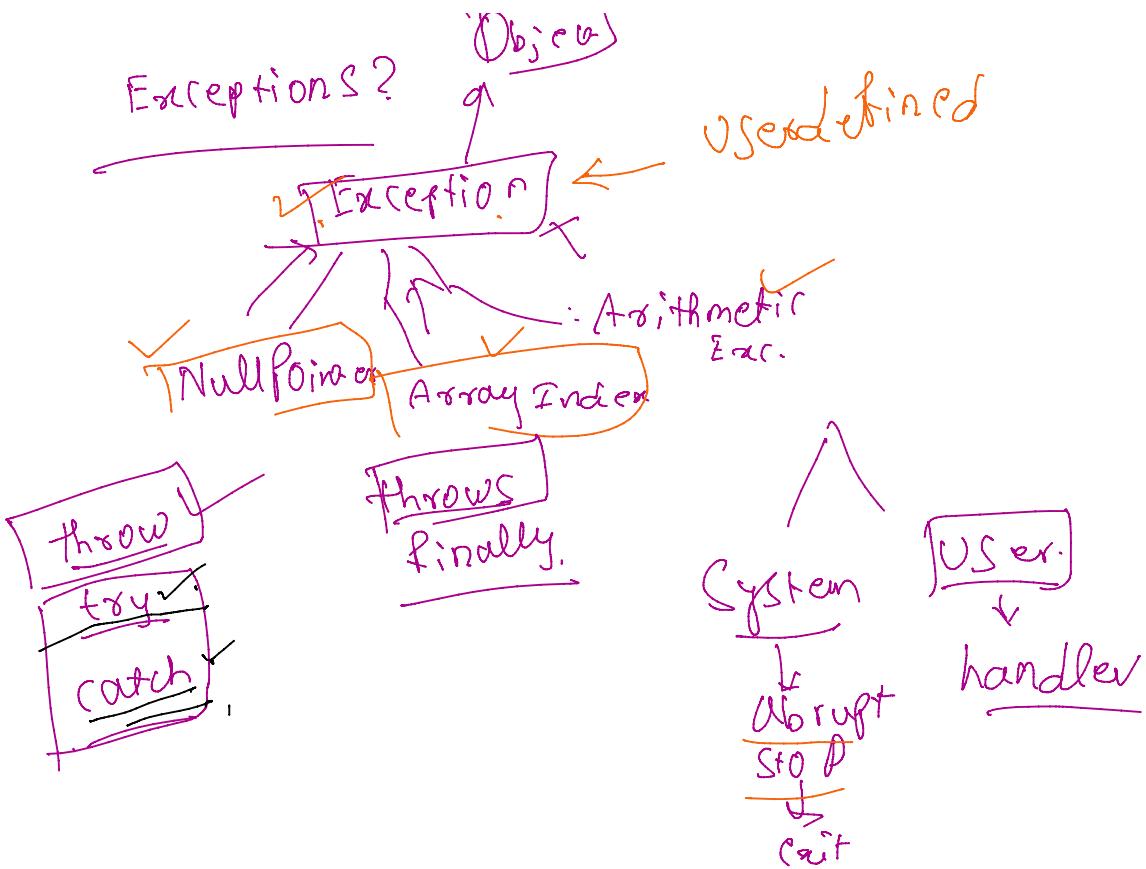
- \* Read integer, float value from user as String, extract the corresponding values (using parseInt / parseFloat) & display the value multiplied by  $x_2;$
- \* Add / Subtract / mul / Divide numbers using wrapper class objects & Identify Boxing & unboxing mechanisms.

### Exception Handling:

- \* String s = null;  
sop(s.length());
  - \* int [] array = new int[4];  
array[4] = 10;
  - \* int a = 3, b = 0;  
int c = a / b;
- Exceptions? Object a

1 line





try {  
 if code which may cause throw  
 Nullpointer Exception or Arithmetic Exception ...  
} catch (NullpointerException e) {  
 what to do when exception occurs.  
}  
catch (ArithmaticException)  
catch (ArrayIndexOutOfBoundsException)

catch (ArithmaticException e)

catch (Exception e)

detail

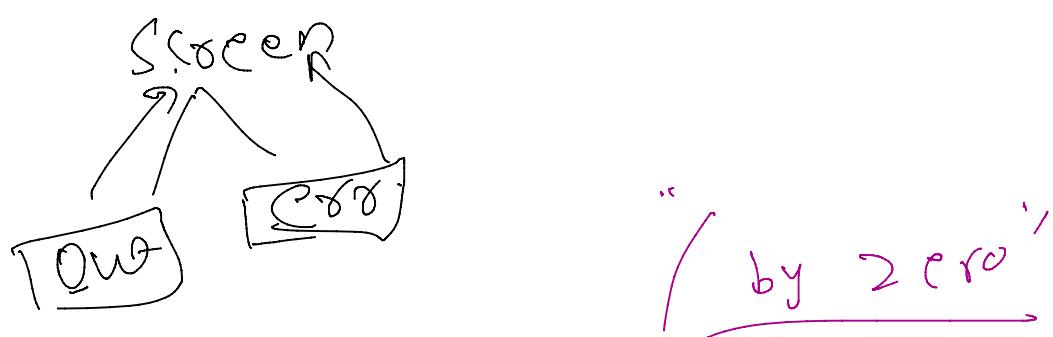
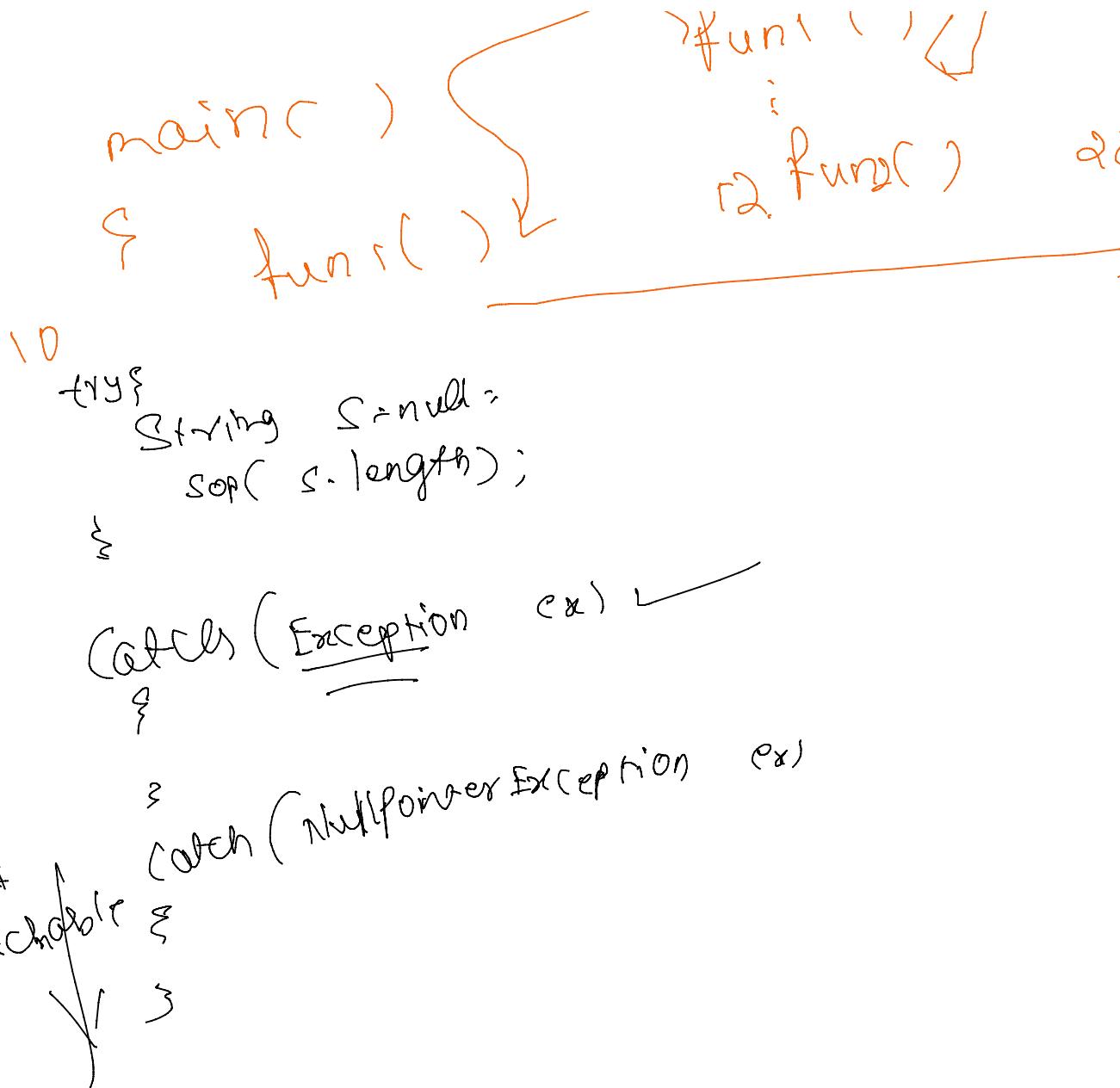
fun( )

ception  
ject

o f.

}  
fun2()

- - -



int  $a = b/c;$

if ( $c == 0$ )

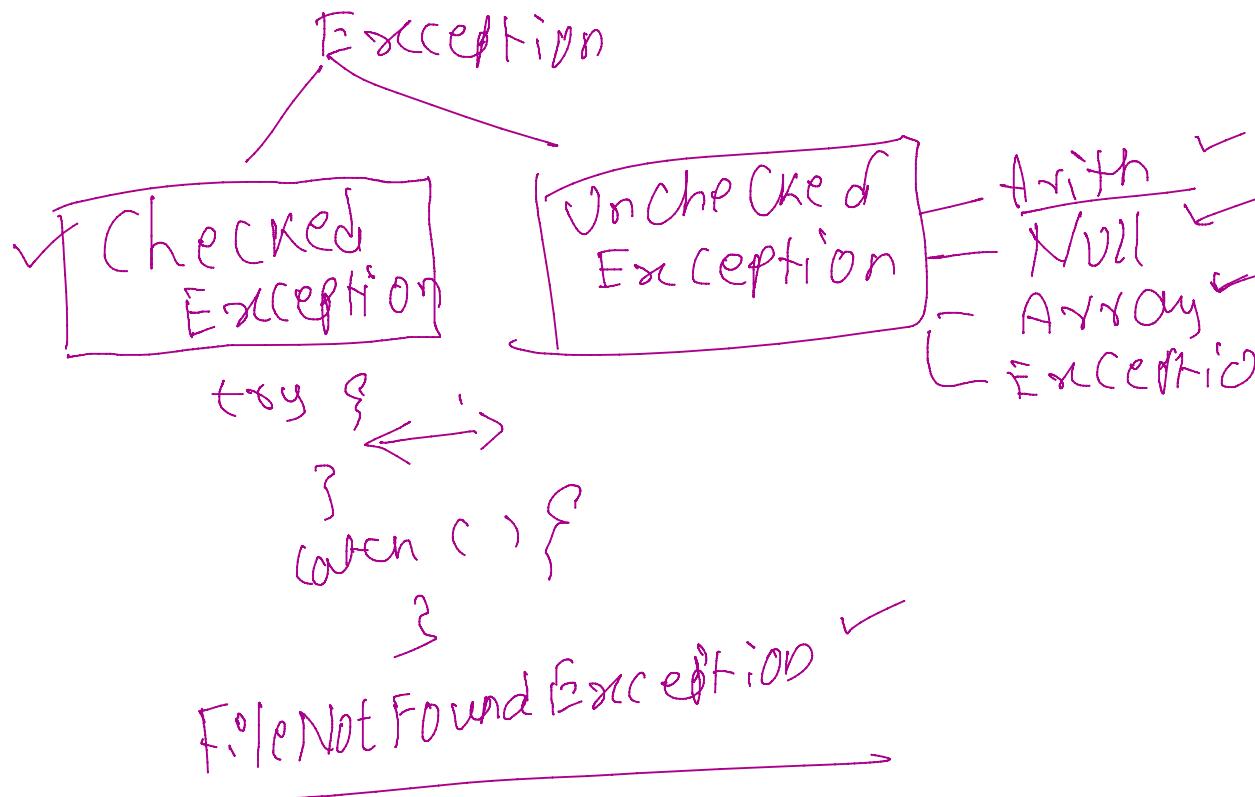
..... ArithmeticException

② Exception



`if ( c == 0 )` new ArithmeticException ( " Divide by zero " )

throw



throws :

```

ps v main ( )
{
    try {
        readFromFile (filename);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```
    throws FileNotFoundException
  
```

↗  
—  
↓ ↙

↙  
of found fix

void (readFromFile)(String filename) run  
file file = new file(filename);  
fileInputStream fis = new  
fis(file);  
checked exception



