

Custom Exception classes

in a day For ATM withdrawal

40,000

withdraw → add to total withdraw
if (total > 40000)

throw new
[WithdrawLimitExceededException]
(') :

checked exception ✓

WithdrawLimitExceededException extends
Exception {

?

Custom Exception Demo

bal → 50,000

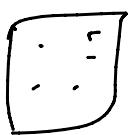
com. forsy harris. model
↳ Person

. driver
↳ Demo

. exception
↳ Exception class.

System

out Text
↓ general
error red.



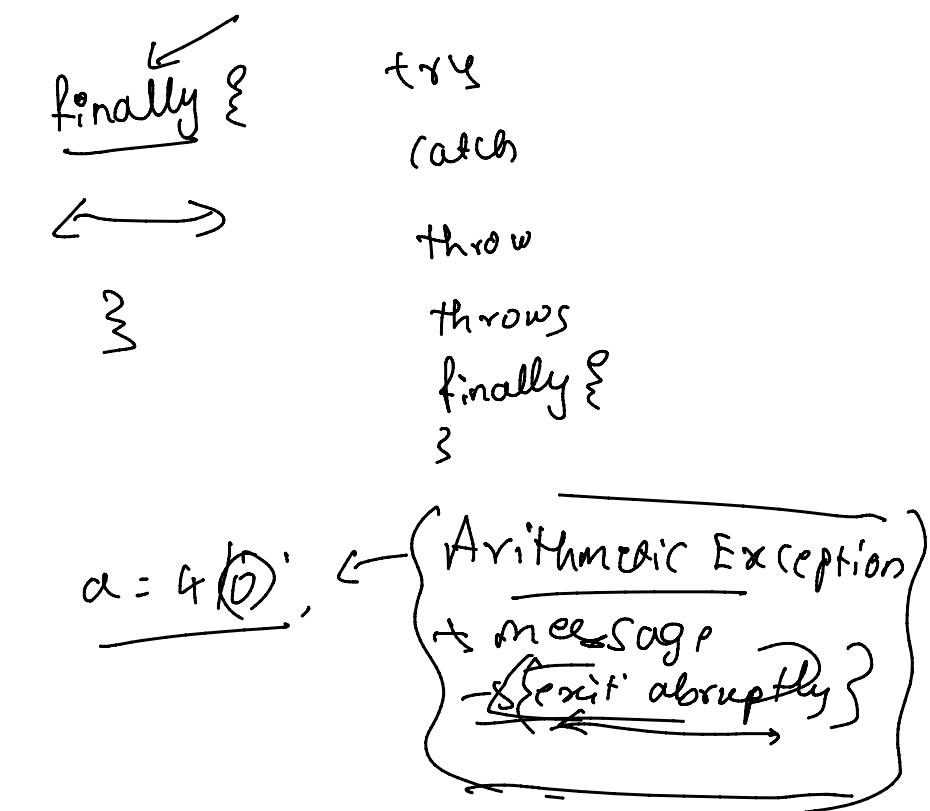
↑
general
low

→ γεγ.

11

Withdrawal Lim.

l) Insufficient Funds Exception



finally {

! →

2

Exception raised → disp.msg ->

finally code -> exit.

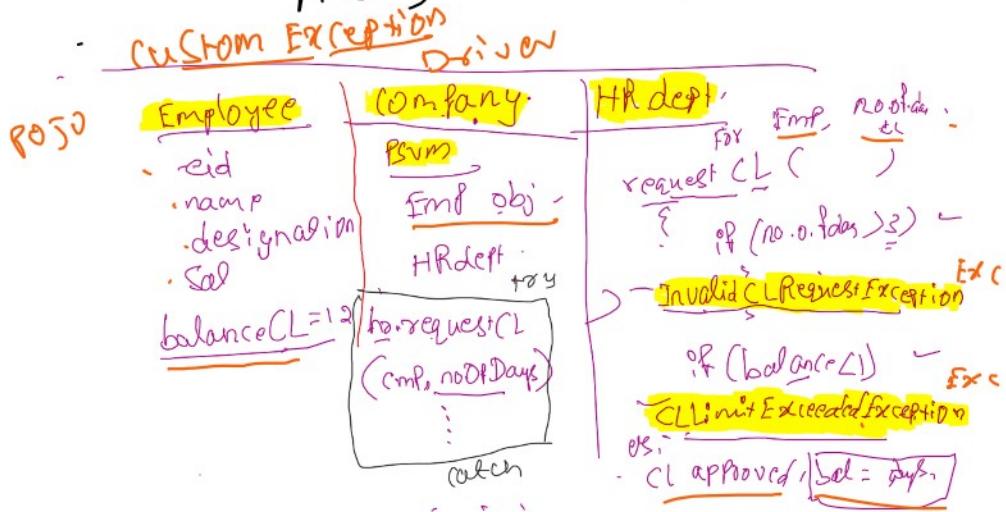
try }

2

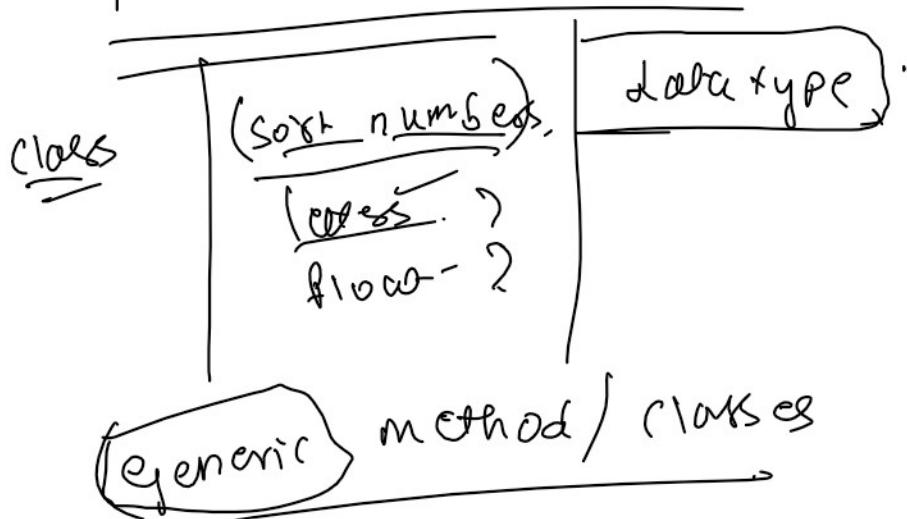
→ finally {

cath.

A hand-drawn diagram illustrating exception handling. On the left, a brace groups the words "try" and "catch". An arrow points from this group to another brace on the right, which groups the words "try" and "finally".



Generics & collections!



to point an object

class Point<T>

{
 T obj;

generic Type

Data Type X

(10x)
Emp.
Obj.
Print(T obj)
this.obj = Obj'
3
 public T getObj()
return this.obj
3

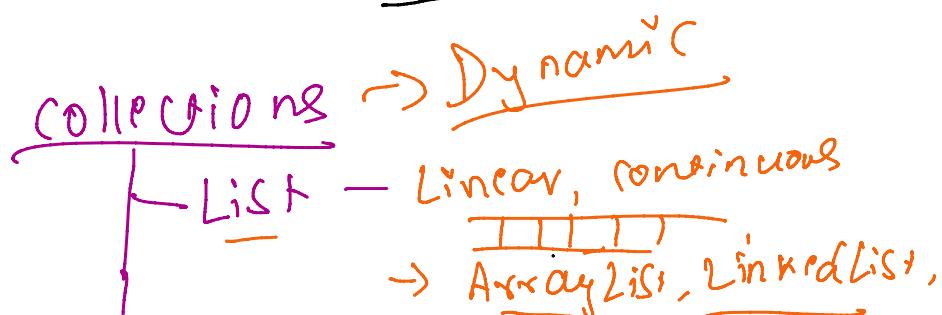
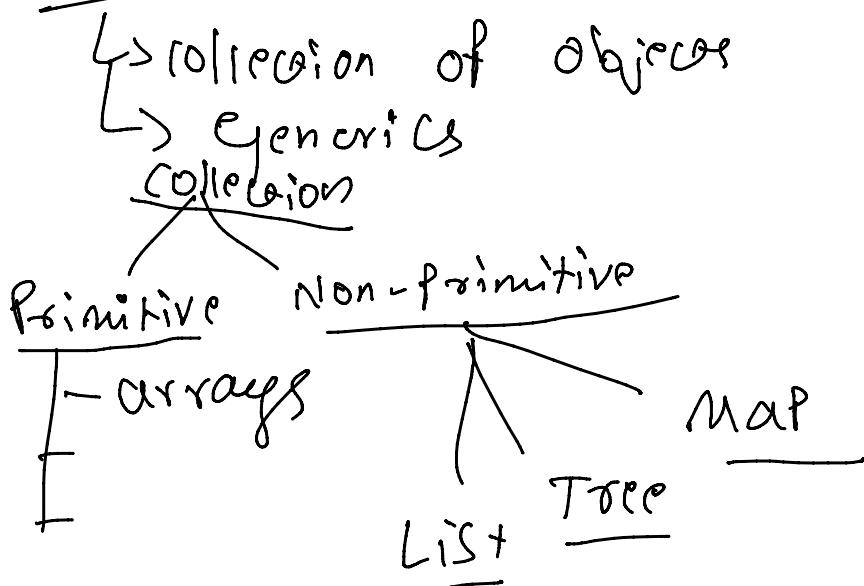
GenericPrint<Integer,Float>

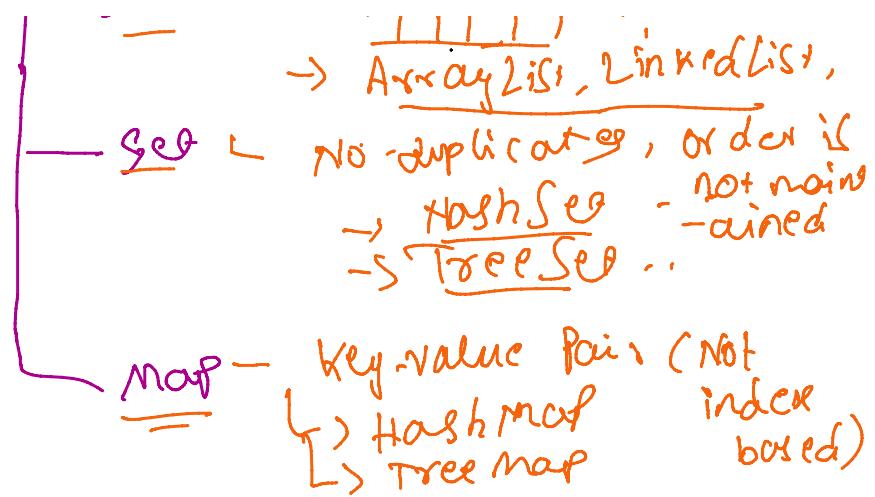
```

package com.torryharris.model;
public class GenericPrint<T> {
    private T obj;
    public GenericPrint(T obj) {
        this.obj = obj;
    }
    public T getObj() {
        return obj;
    }
}

package com.torryharris.driver;
import com.torryharris.model.GenericPrint;
public class GenericsDemo {
    public static void main(String[] args) {
        GenericPrint<Integer> iPrint = new GenericPrint<>(obj: 10);
        System.out.println(iPrint.getObj());
        GenericPrint<String> strPrint = new GenericPrint<>(obj: "Demo");
        System.out.println(strPrint.getObj());
        GenericPrint<Float> fPrint = new GenericPrint<>(obj: 3.2f);
        System.out.println(fPrint.getObj());
    }
}
    
```

Collections in Java :





List → ArrayList
 → LinkedList

