1) Friday → 02/08 ⇒ Java Adv 3 ⇒ Thursday 01/08

2)

19/08 ⇒ Mon

10/08 ⇒ Saturday

12/08 ⇒ Monday

14/08 ⇒ Wed

16/08 ⇒ Friday

000 ⇒ 13/08 to 19/08

Multi-threading

- → Static + multi-threaded
- → deadlock
- → concurrent data structure
- → more deep concepts — [Program Counter]
- → synchronised method

Generics

→ Store the data for a student — [name, id]

class Student {

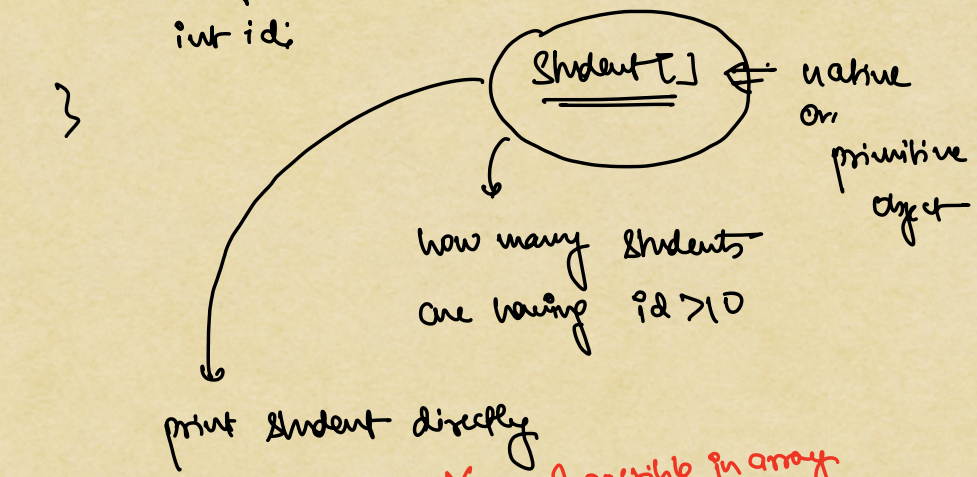
String name;

int id;

}

int[]

boolean[]



class StudentList {

Student[] Students;

int size;

print()

getMoreThanId(id)

sort()

}

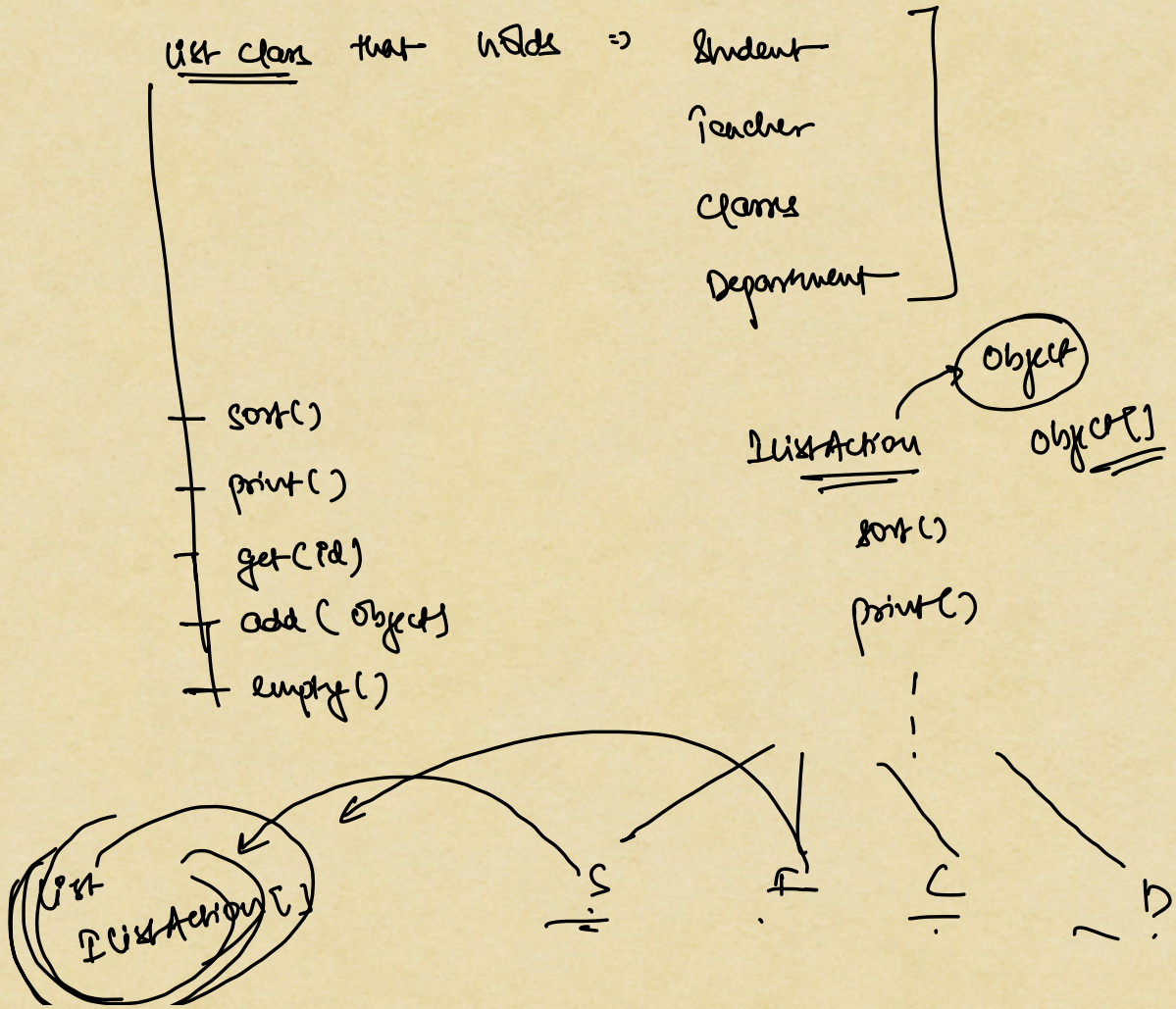
StudentList.print();

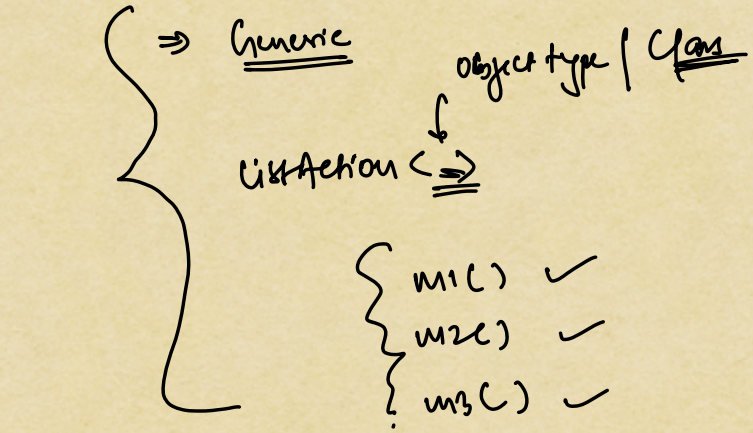
⇒ List of Teacher [name, id, dept]

- print
- sort
- getTeacherByDept

TeacherList {

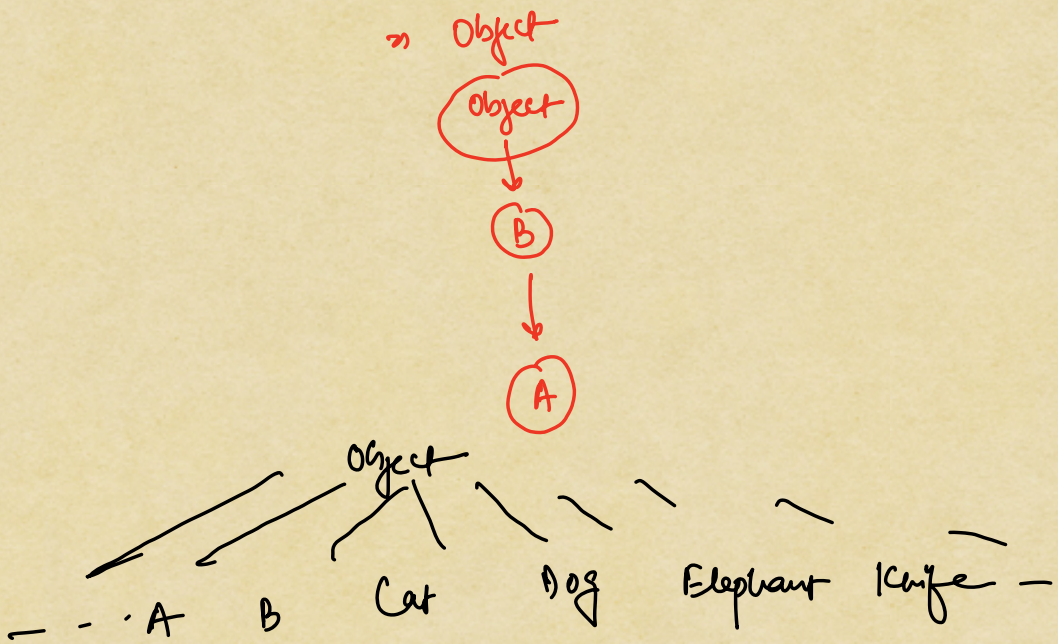
}





* what methods every object in java will have?

Class which is parent of classes in java

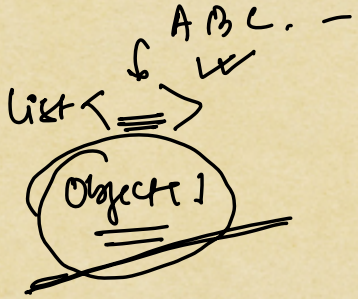


A
↓
B

A a = new B() ← upcasting

A[] array = new A[1];

array[0] = new B();



Object[] arr = new Object[]

↓

arr[0] = new A();
arr[1] = new B();
arr[2] = new C();

int arr :: 1 2 3 4 5 6 7

boolean arr :: t f t f

⇒ Generic

↓

allow to store any type

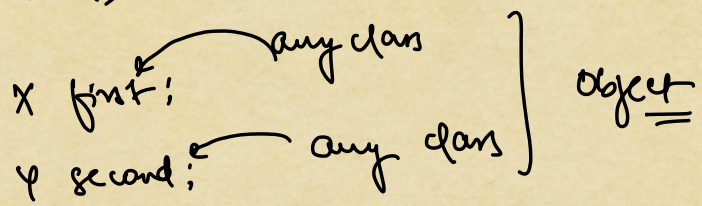
↑

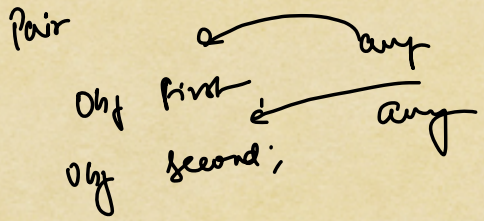
before initialise

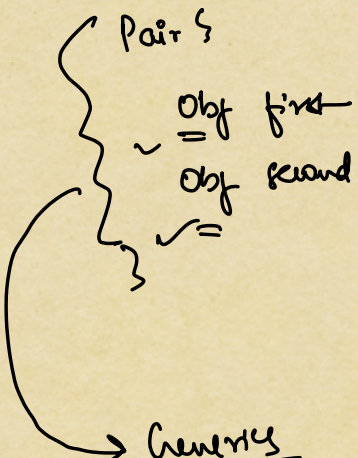
↓

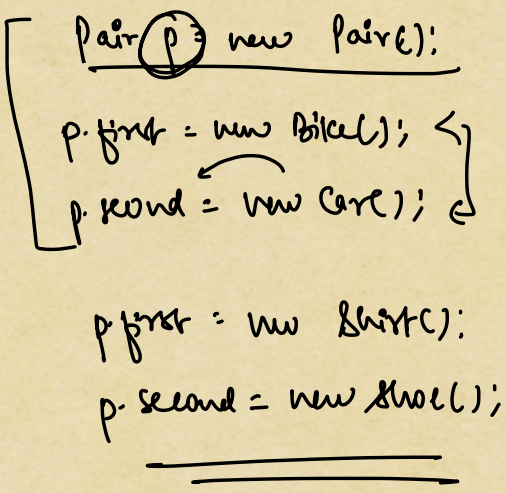
after initialisation ⇒ type is fixed

Pair <X, Y>


any class
X first;
Y second;
any class
Object


Pair
any
Obj first
any
Obj second;


Pair {
Obj first
Obj second
Generics
allowed to tighten
the type casted


Pair p = new Pair();
p.first = new Bike();
p.second = new Car();
p.first = new Shirt();
p.second = new Shoe();

Pair <X, Y> {

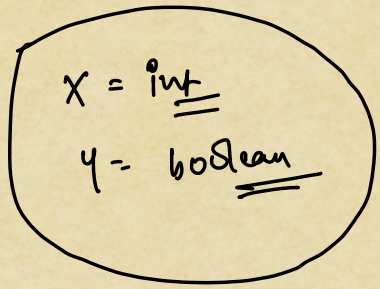
Integer X first;

Y second;

any class

any class

}



Object → ~~primitive~~

Wrapper classes

int → Integer

boolean → Boolean

char → Character

long → Long

!

Integer {

int x;

}

Pair <X, Y> {

X first;

Y second;

}

Pair <Integer, Boolean> p = new Pair<>();

p.first = ~~new Car();~~

Pair<Car, Bike> p1 = new Pair();

↓

Car, Bike

Raw Data Types