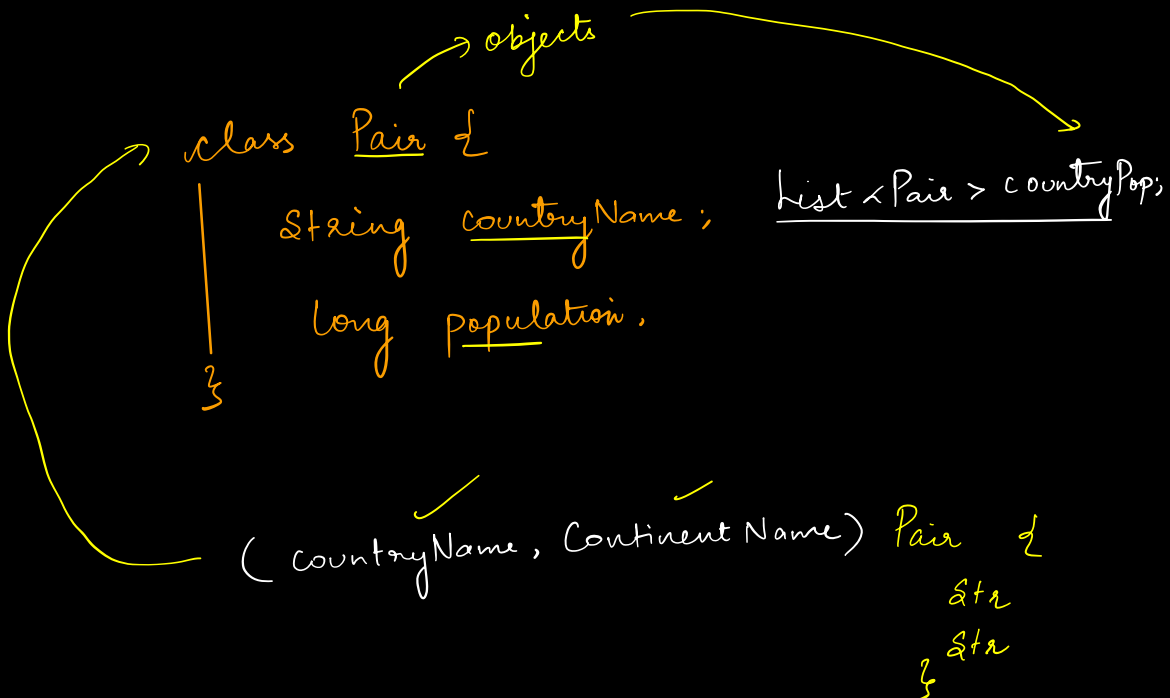
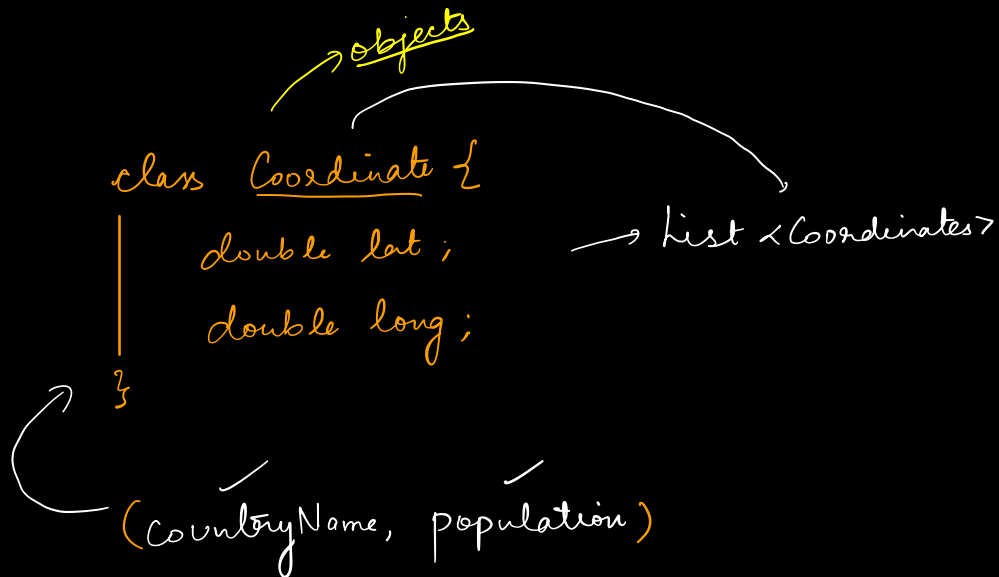


Today's Agenda :-

Generics

2D array

list of coordinates = { (- -),
 ↓
 (lat, long) (- -),
 (- -) }



```
class Pair {
```

```
    { double lat; }  
    { double long; }  
}
```

```
Pair p = new Pair();
```

~~String, long~~ x

We are restricting the datatype

Make this Pair a generic one?

✓ Object class in Java

↳ class in Java which is
parent of every other
class in Java.

```
class Pair {
```

```
    Object first; } can accept any data type  
    Object second; } values.  
}
```

→ (Country-Name, population) → class

```
Pair p1 = new Pair();
```

```
p1.first = "America"
```

```
p1.second = 21,000,000 ✓
```

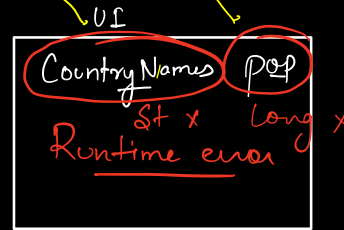
```
p1.second = "India" } Problem?
```

Integer vs int? x
Read.

Pl. first = "21,532"

↳ Run time error

No Compile-time
type check



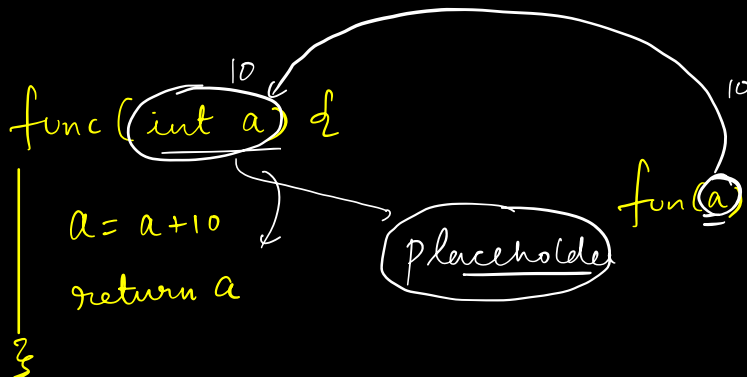
"Fail fast"

To ensure compile time check & as well as
keep data types generics



Generics
↓

Parameterized Data types.



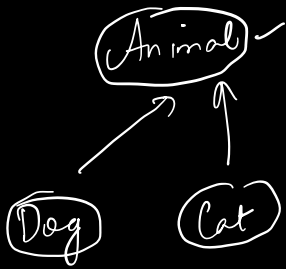
```
class Pair < A, B > {  
    A first  
    B second  
}
```

```
Pair < Integer, String > p = new Pair <>();
```

p1. first = "xyz" X

p1. first = 25 ✓

} Generics
Applies a
Compile time check



doSomething (List < Animal > animals) {

}

animals.add(new Dog());
 animals.add(new Cat());
 animals.add(new Animal());

} ~~==~~

Animal x = new Dog(); ✓

List < Dog > dogs = { ... }

doSomething(dogs)

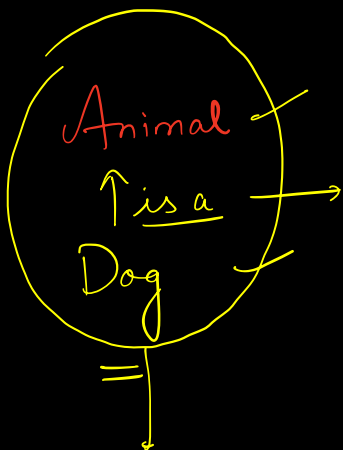
List < Animals >

=

new List < Dogs > ();

new Dog();
 new Cat();
 new Animal();

{ new Dog();
 new Cat();
 new Animal(); } X



Animal (animal) = new Dog()

List < Animal > ✓
↑ X

List < Dog > ✓

List < Animal > = ~~new List < Dog >~~
↳ new Cat(),
new Dog(),