OPPs-2

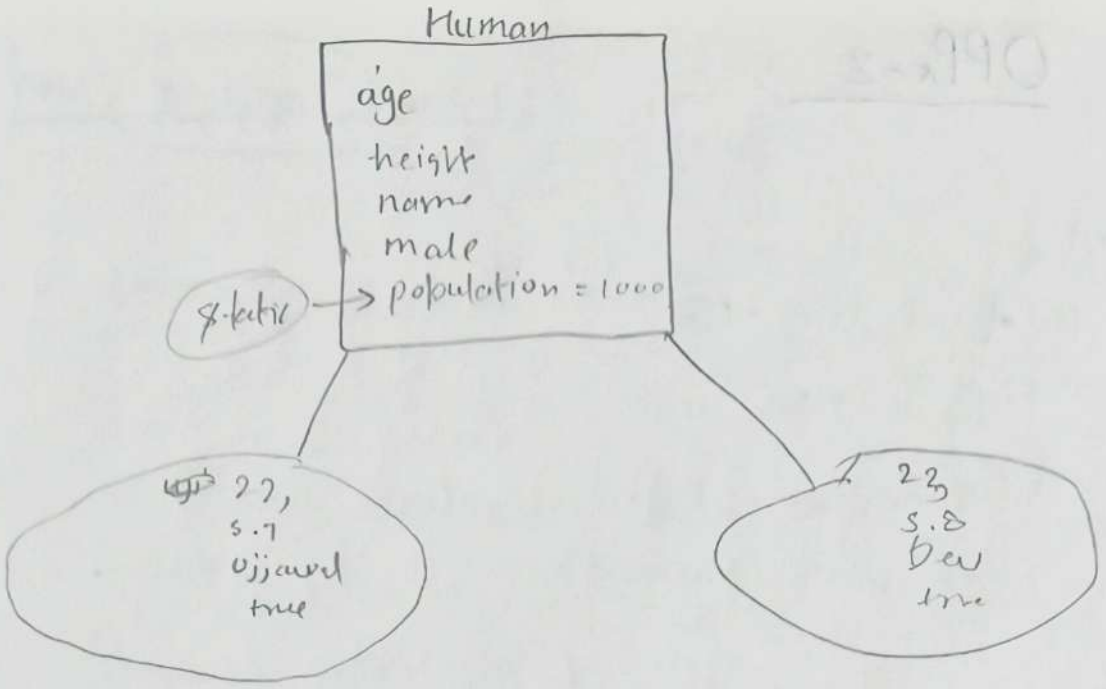
```
Class A {
    final int num = 10;
    String name;
    A (string name) {
        this.name = name;
    }
}
```

```
>> main() ──> {
        A obj = new A("ABC");
        System.out.println(obj);
```

>> //somehased code - //we can override

Packages - container for classes → just file managment stuff.

Static - Such properties which are not directly related to the obj, those are known as static.



So basically -
in constructor →

```
public Human (   ) {        → passe Argum

    this.age = age
    this.height = height
    this.nam = na
    this.male = male
Human.population += popula
```

↳ we can use "this" too but
when we use this.population += 1, we basically
try obj1.population += 1
but there is nothing called population to obj1
so we check for the class, does class have value
of population & it's static, it's common for all,
it will be updated.

& for next object we'll go for
same

So basically we class name to access the static values & methods because it independent of object.

SOUT(Human.population);

Non-static members inside a static -

Ex - public static void main (String[] args) {

}

why it's static → to run it first without creating obj.

Static method only can access the static data.

Because when try to call any non static function in static main() → it will give error

why → logically you can't use any non static value without the object & greet is just a function of class "Main"

```
public class Main{
  main( ) {
    greet();
  }
  void greet ( ) {
    SOUT ("Hi Ujjawal");
  }
}
```

so, greet(); → error

but Main obj = new Main();
obj.greet(); → work ✓

this keyword inside static -

```
public class Innerclasses {
    static class Test {
        String name;
        public Test( String name) {
            this. name = name;
        }
    }
    public static void main(String[] args) {
        Test a = new Test (" Ujju");
        Test b = new Test (" Dev");
```

} this will give error if we don't put static before the Test class

why

We can't put any static thing inside the non-static

if not -
then, it will need to create instance of outer class ("Innerclass") ⟶ so for that we need to use static before test -

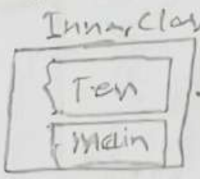
---

What if we put outsid the inner class —
then As usually we don't get error as we can make $\boxed{\text{Test a = new Test ( )}}$ object —

So basically, when we run the program -

```
>> Ujju
>> Dev
```

But how can static be dependent on object?
⟶ Because of scope — where the static is



⟶ so The main() can have obj of Test —

Singleton Class - This mean
A class of which you can create one object -

Basically -
Stop them using constructor -

So make it private -

```
public class Singleton{
    privat Singleton (){
    }
    private static Singleton instance;    // Basically created a
                                             private object whic
    public static Singleton getInstance(){   cann't be access
      //check -                              by any call from
      if (instance == null){                 diff class
          instance = new singleton();
      }
      return instance;
```

```
public class Main{
    public static void main (String [] args){
        singleton obj1 = Singleton.getInstance(); → will create
                                                     an object
        Singleton obj2 = Singleton.getInstance(); → point to sa
                                                     instance
```

So, Basically we created a privat constructor & a private instane that will be accessed by by getInstance() & then we put if statement to check we already have instance or not.