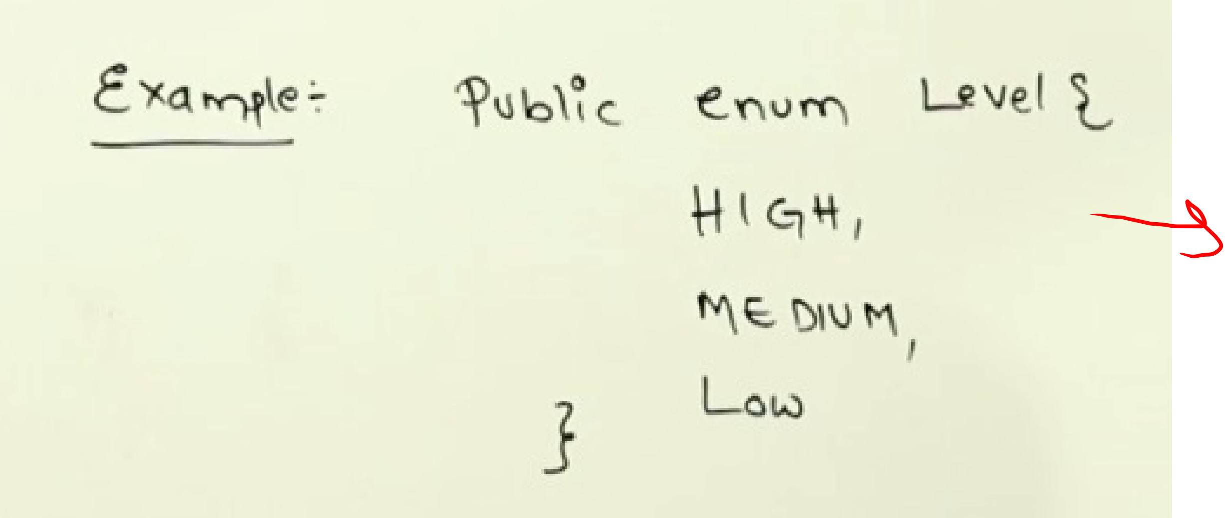
Enum → Short form of Enumeration

Variable has value which can take some values

i.e. Constants like Week {MON, TUE, WED, THRUSH, FRI, SAT, SUN} It is a special type of class

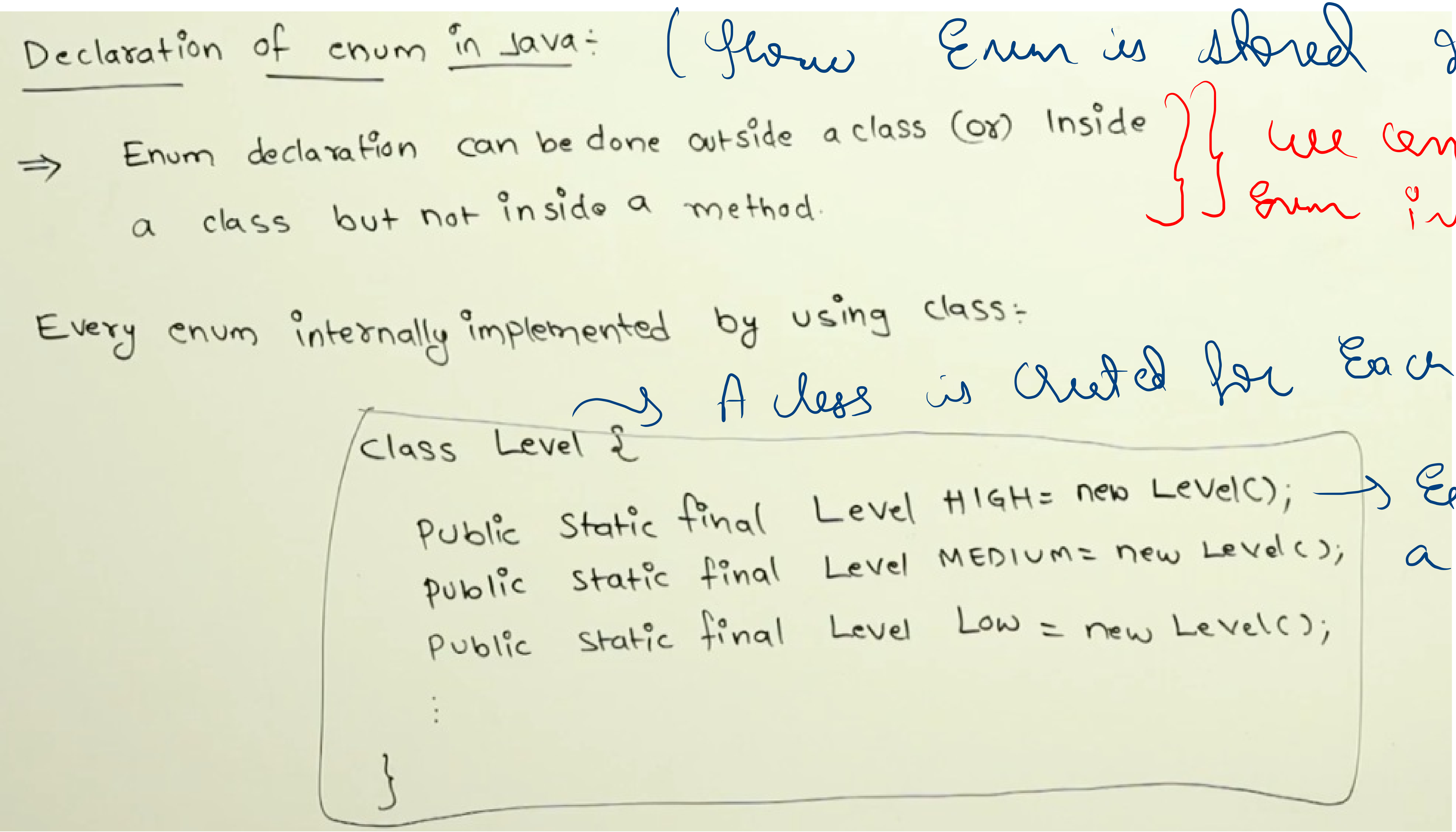


Example:

```
Public enum Level {
        HIGH,
        MEDIUM,
        Low
}
```

→ To Refer

Level l = Level.HIGH;



Declaration of enum in Java:

(How Enum is stored internally)

⇒ Enum declaration can be done outside a class (or) Inside a class but not inside a method.

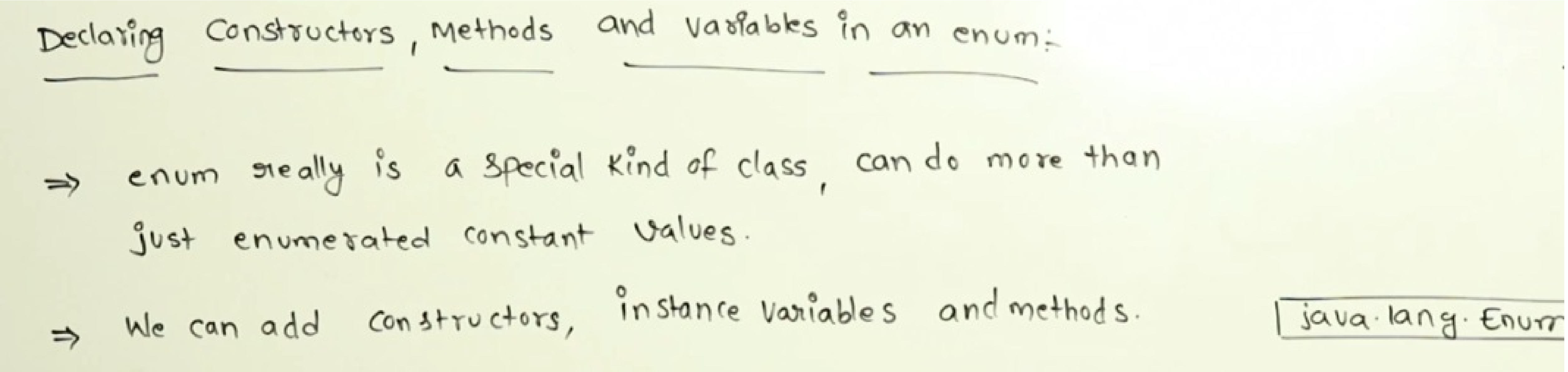
we cant declare Enum inside a method

Every enum internally implemented by using class:

A class is Created for Each Enum

```
class Level {
    Public Static final Level HIGH= new Level();
    public static final Level MEDIUM= new Level();
    Public static final Level Low = new Level();
    :
}
```

Each value is a Object of that class



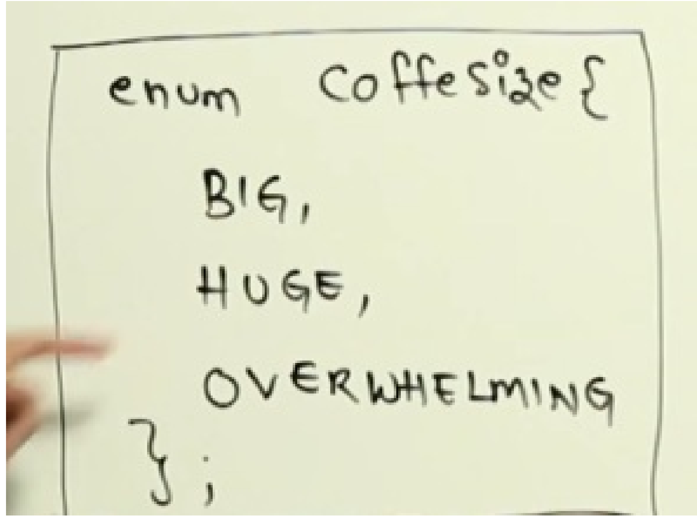
Declaring Constructors, Methods and variables in an enum:

⇒ enum really is a special kind of class, can do more than just enumerated constant values.

⇒ We can add constructors, instance variables and methods.

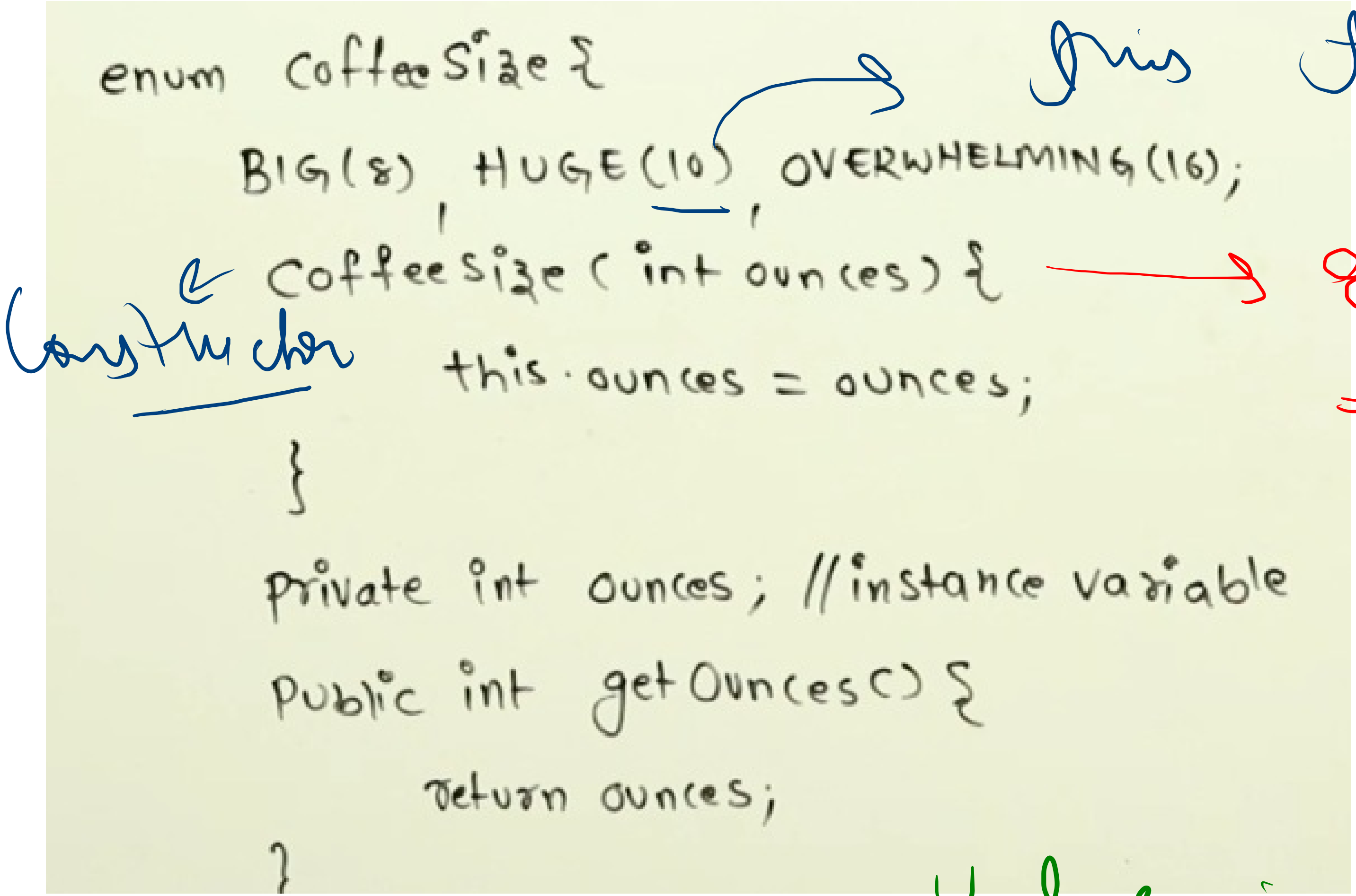
java.lang.Enum

So Enum can hold constructor, method & so on



```
enum CoffeSize{
    BIG,
    HUGE,
    OVERWHELMING
};
```

Internally 3 object is created BIG, HUGE & OVERWHELMING act as reference to once object !!



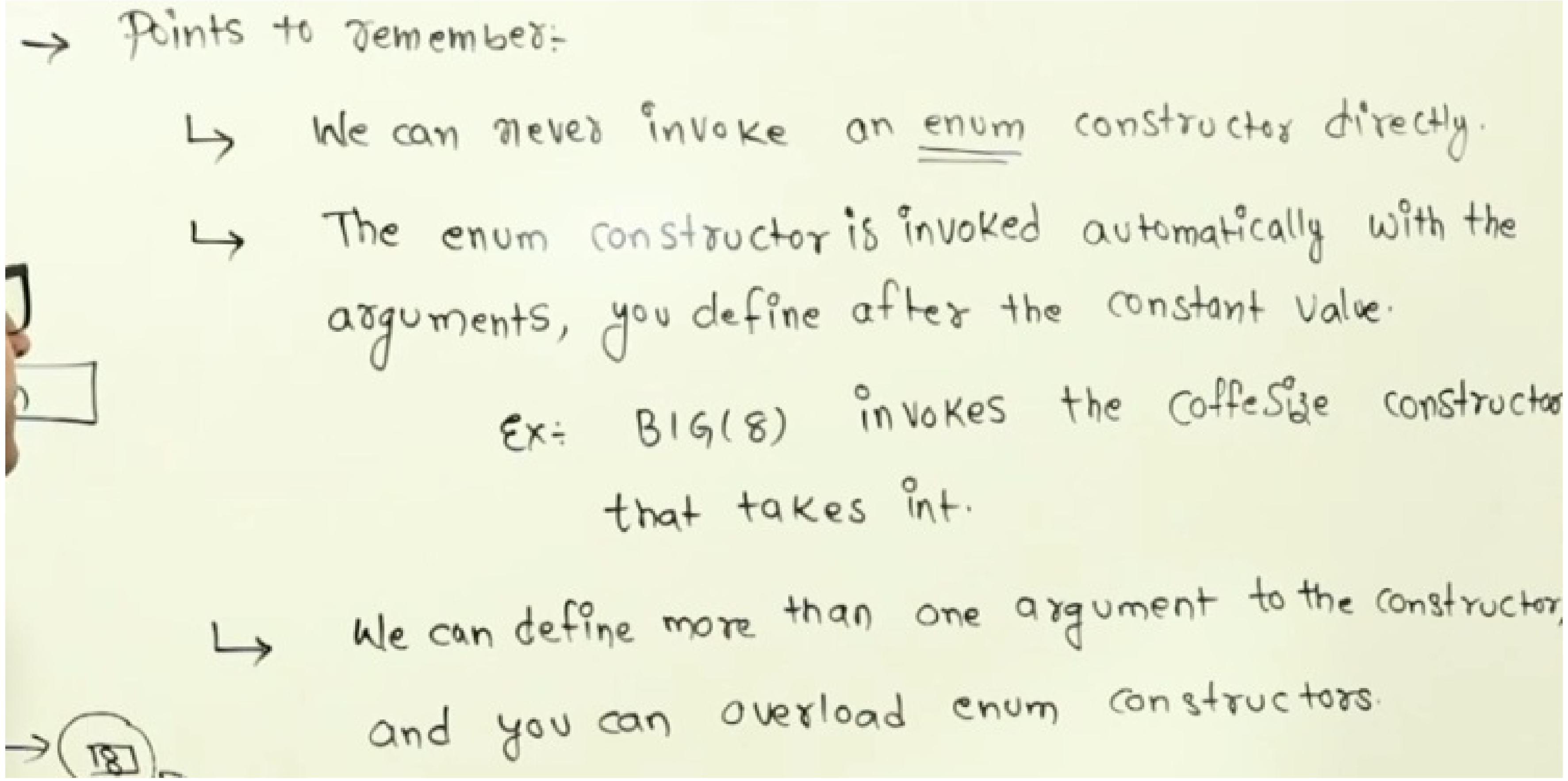
```
enum CoffeeSize {
    BIG(8), HUGE(10), OVERWHELMING(16);
    CoffeeSize(int ounces) {
        this.ounces = ounces;
    }
    private int ounces; //instance variable
    Public int getOunces() {
        return ounces;
    }
}
```

This tells HUGE value is 16

Constructor

Enum constructor is private
→ you can't call it

Now see for each BIG, HUGE & OVERWHELMING, Constructor is called & BIG value is set into ounces & then BIG refers to 8 & HUGE to 10 & OVERWHELMING to 16

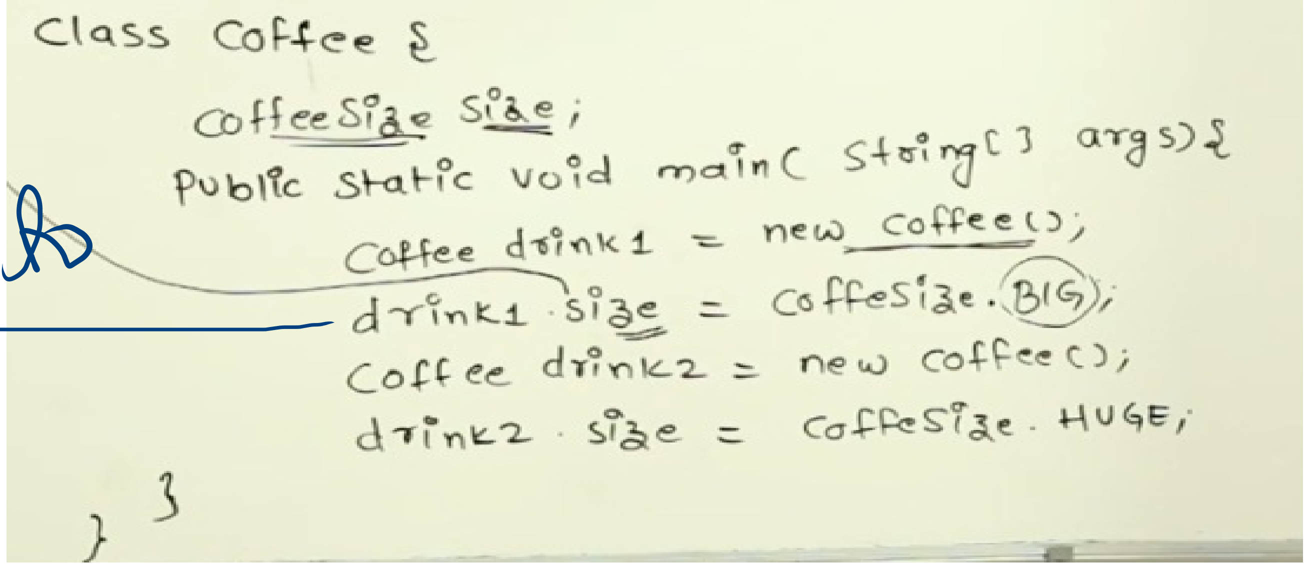


→ Points to remember:-

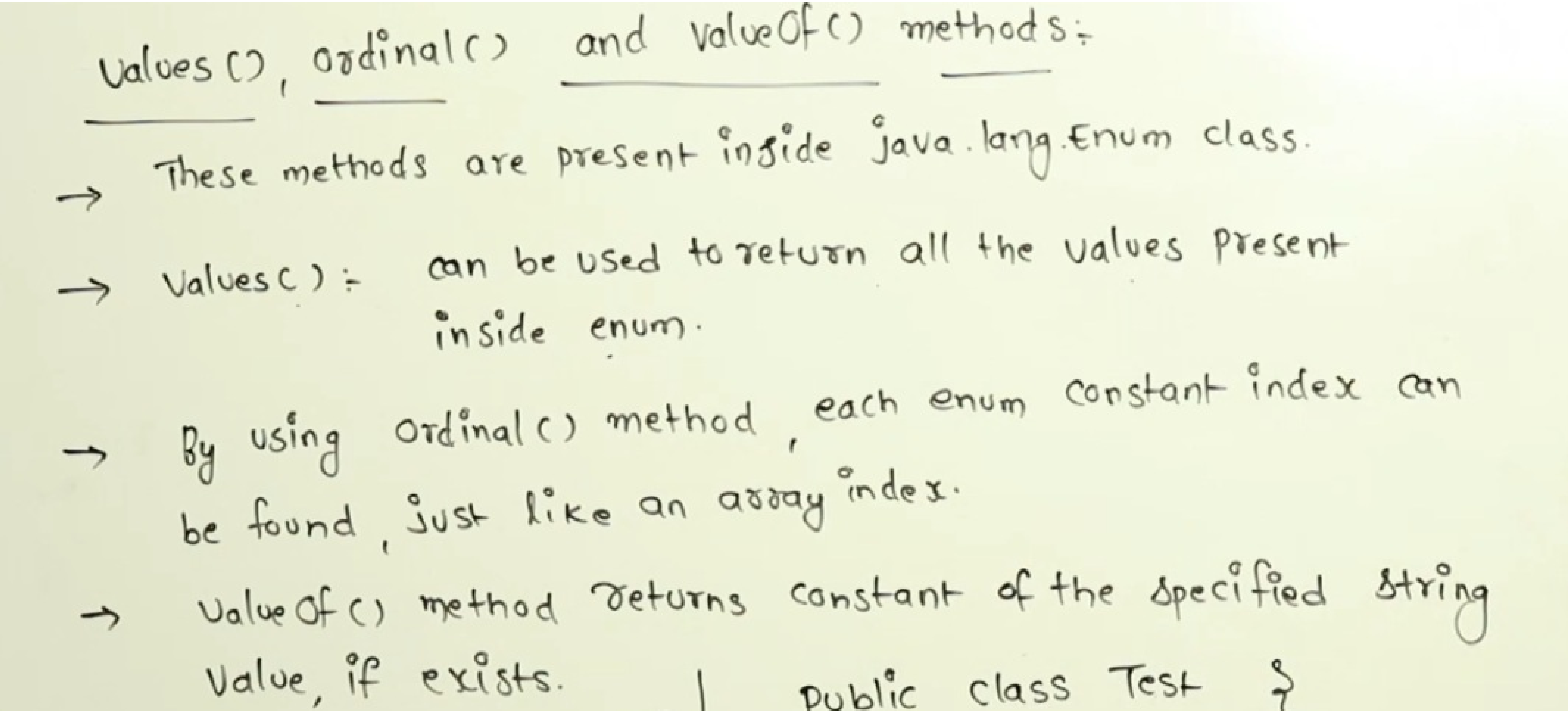
- ↳ We can never invoke an enum constructor directly.
- ↳ The enum constructor is invoked automatically with the arguments, you define after the constant value.

  Ex:- BIG(8) invokes the CoffeSize constructor that takes int.
- ↳ We can define more than one argument to the constructor, and you can overload enum constructors.

Now size refers to that Object which BIG is referring



```
Class Coffee {
    CoffeeSize size;
  Public static void main( String[] args){
      Coffee drink1 = new Coffee();
      drink1.size = CoffeSize.BIG;
      Coffee drink2 = new coffee();
      drink2.size = CoffeSize.HUGE;
  }
}
```



Values(), ordinal() and valueOf() methods:-

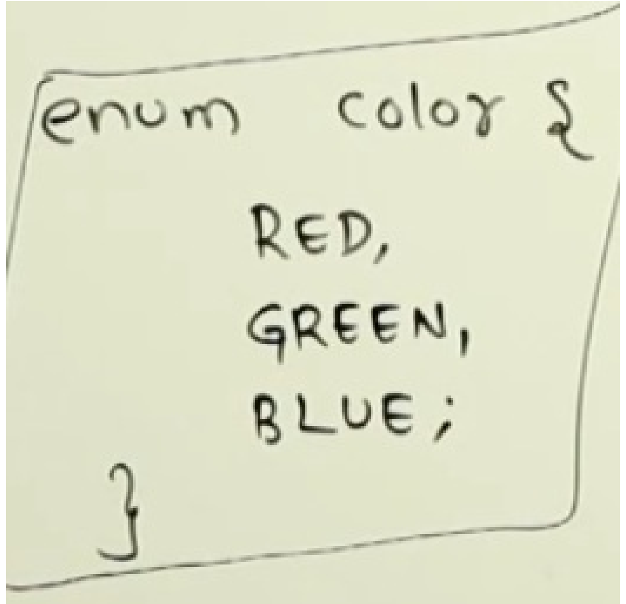
→ These methods are present inside java.lang.Enum class.

→ Values():- can be used to return all the values present inside enum.

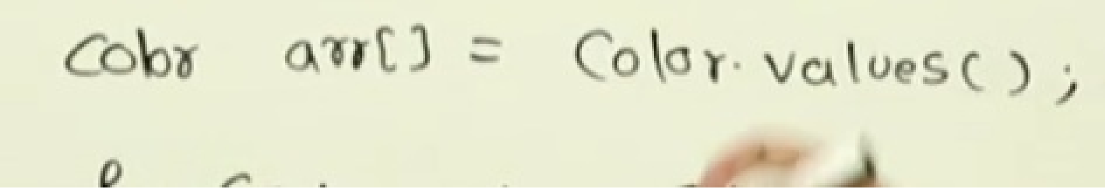
→ By using ordinal() method, each enum constant index can be found, just like an array index.

→ ValueOf() method returns constant of the specified string value, if exists.

```
public class Test {
```

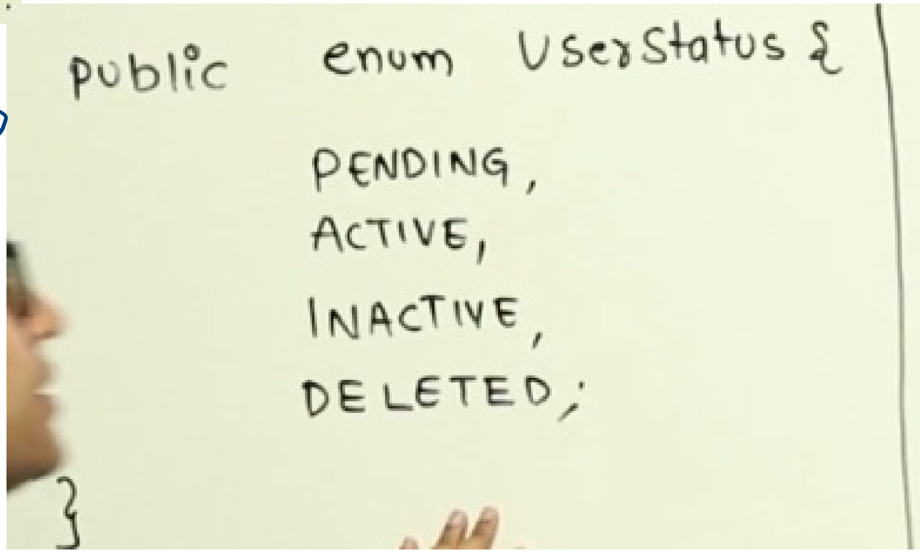


```
enum color {
        RED,
        GREEN,
        BLUE;
}
```

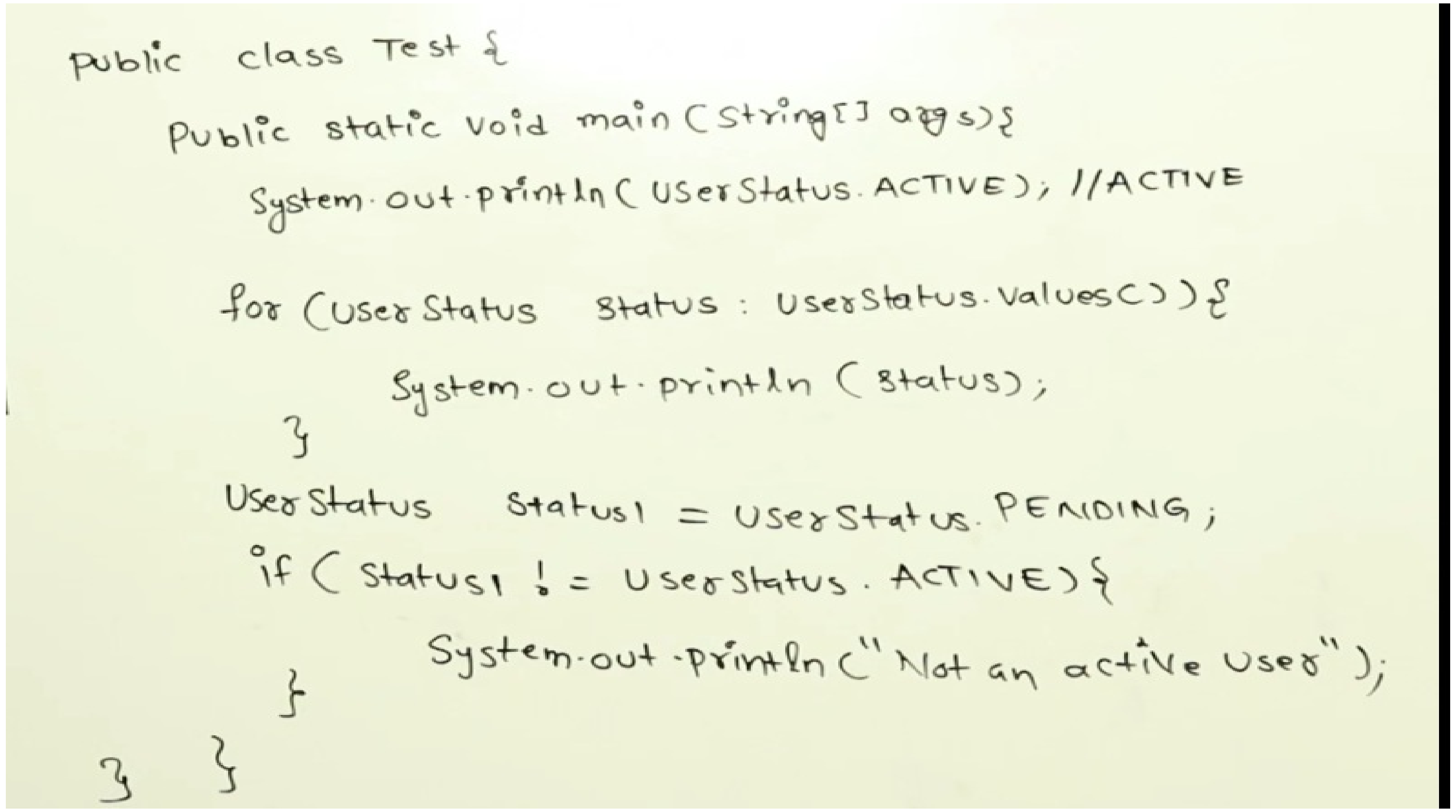


```
Color arr[] = Color.values();
```

Lets See an Example →



```
public enum UserStatus {
        PENDING,
        ACTIVE,
        INACTIVE,
        DELETED;
}
```



```
public class Test {
    Public static void main(String[] args){
        System.out.println(UserStatus.ACTIVE); //ACTIVE

        for (User Status status : UserStatus.values()){
                System.out.println(status);
        }
        User Status  status1 = UserStatus.PENDING;
        if (status1 != UserStatus.ACTIVE){
                System.out.println("Not an active user");
        }
} }
```