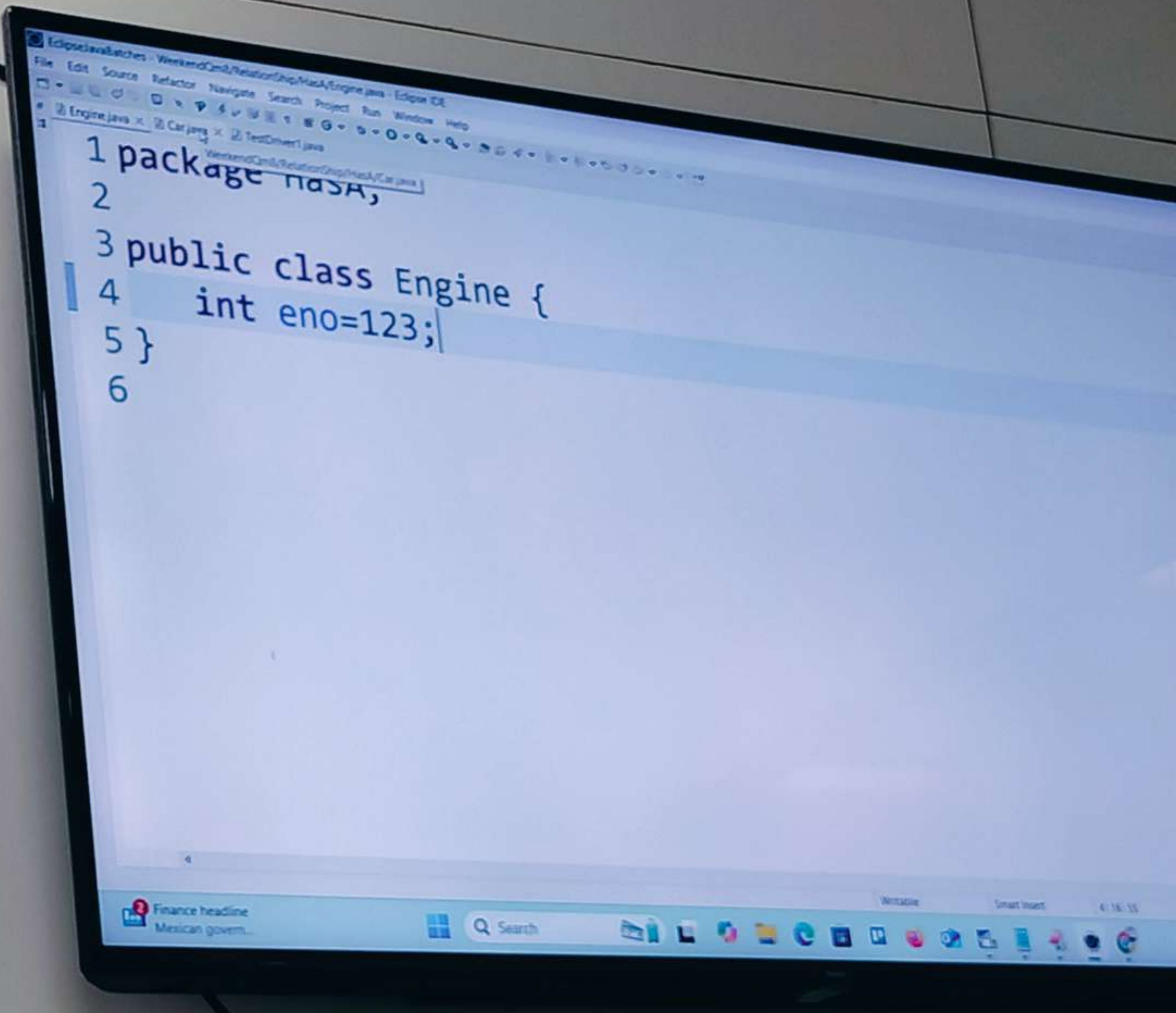
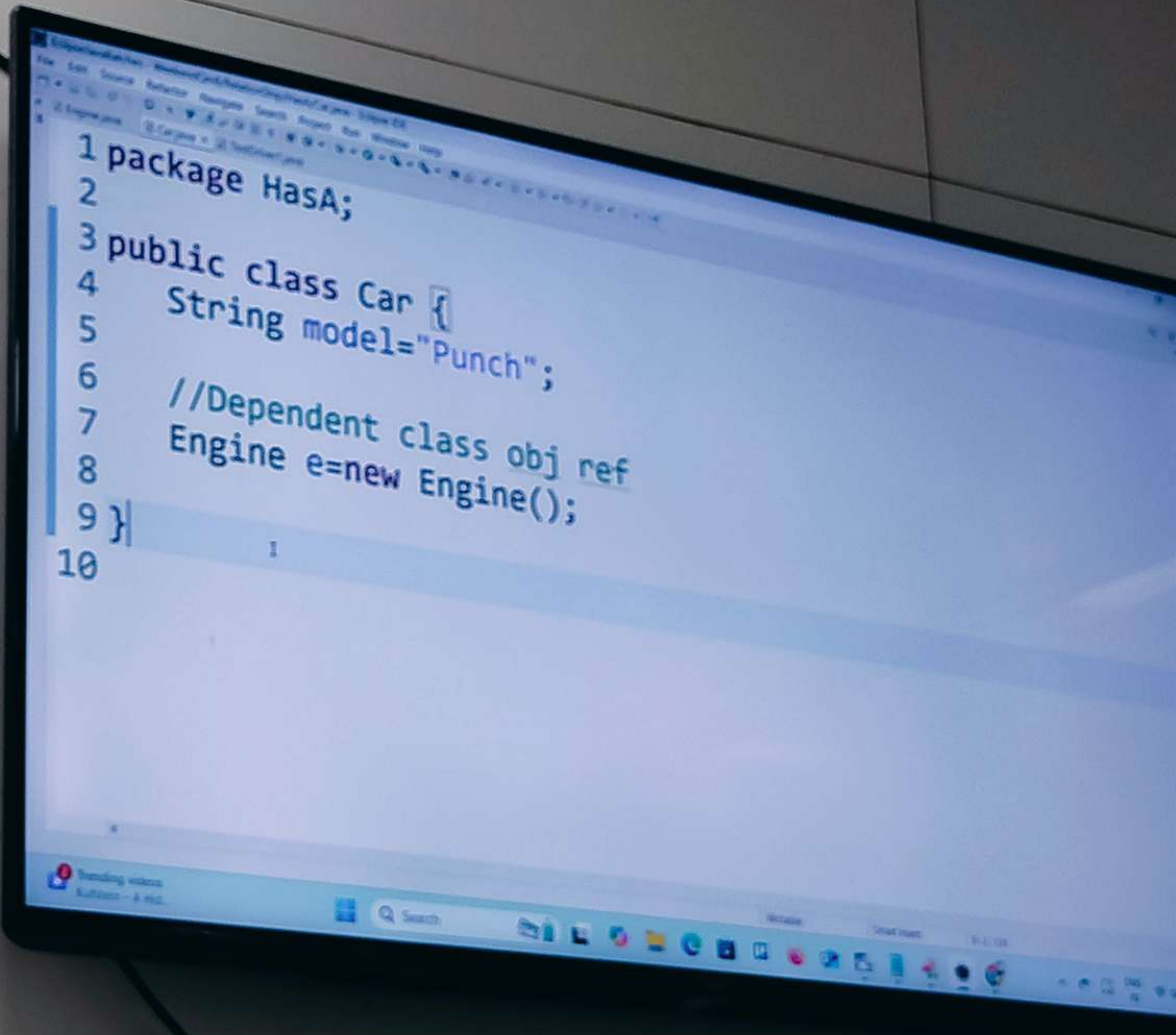
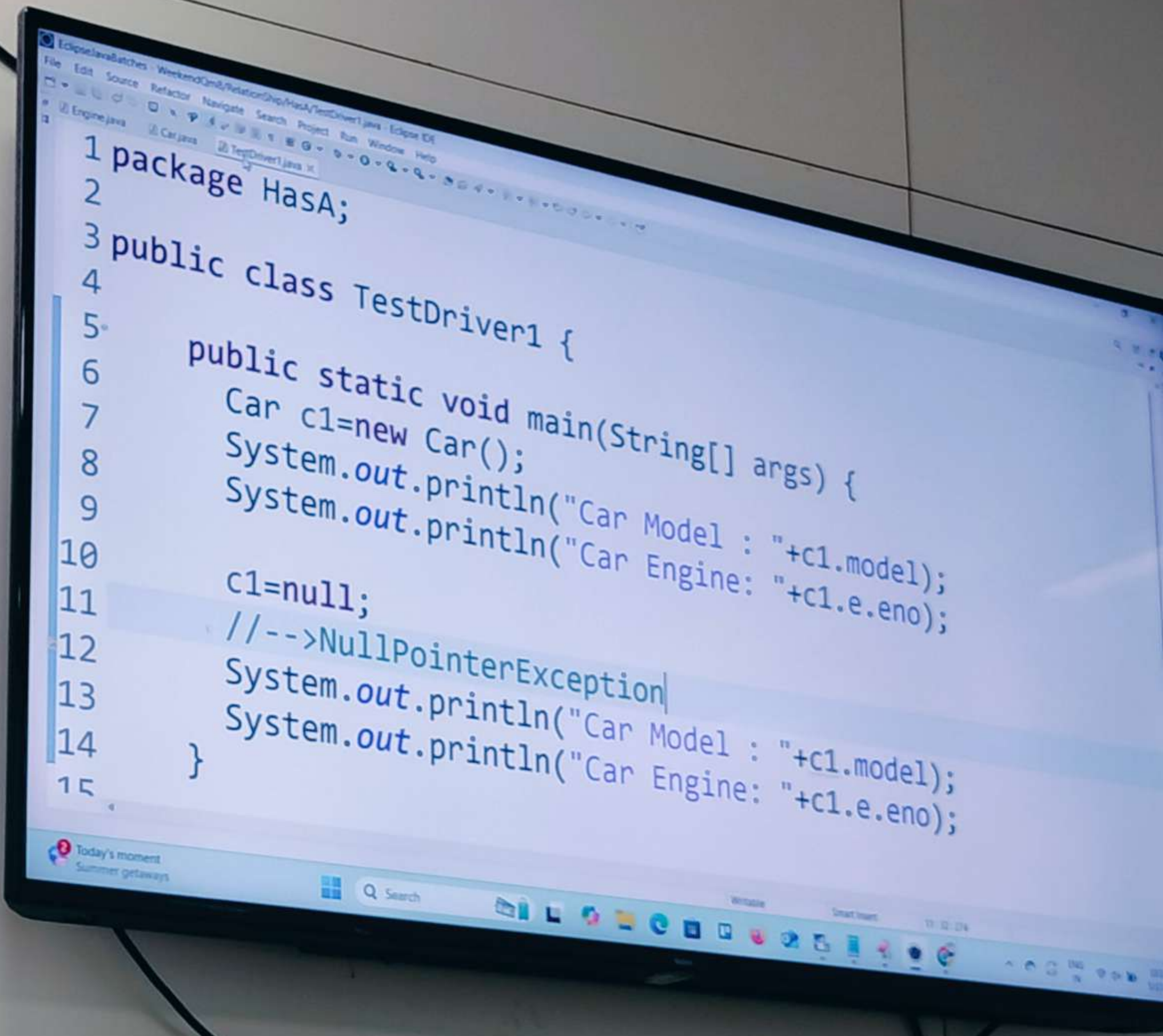

EclipseJavaBatches - WeekendCjm8/RelationShip/HasA/Engine.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Engine.java
Car.java
TestDriver1.java
WeekendCjm8/RelationShip/HasA/Car.java
Finance headline
Mexican govern...
Search
Writable
Smart Insert

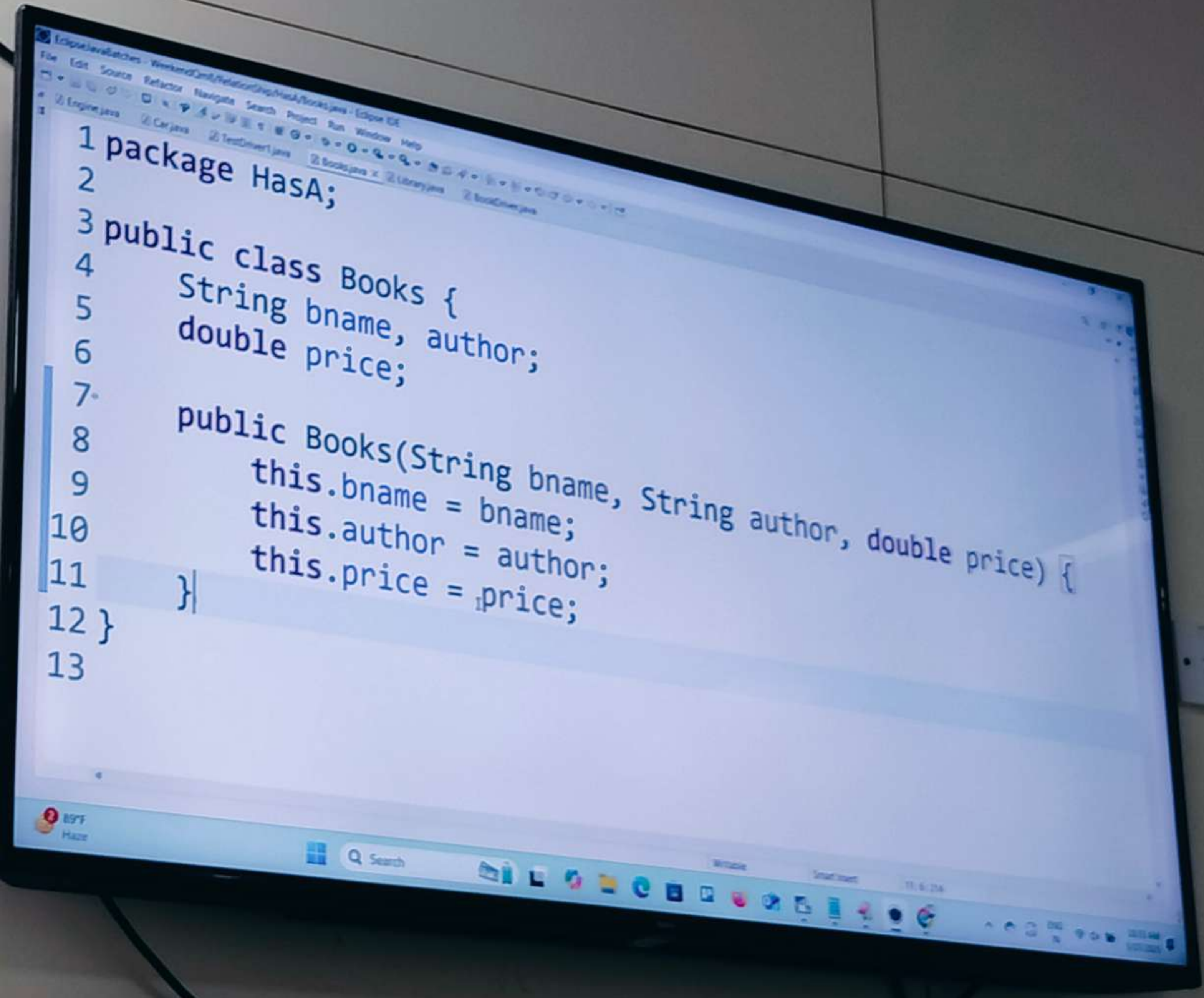
```java
package hasA;

public class Engine {
    int eno=123;
}

```



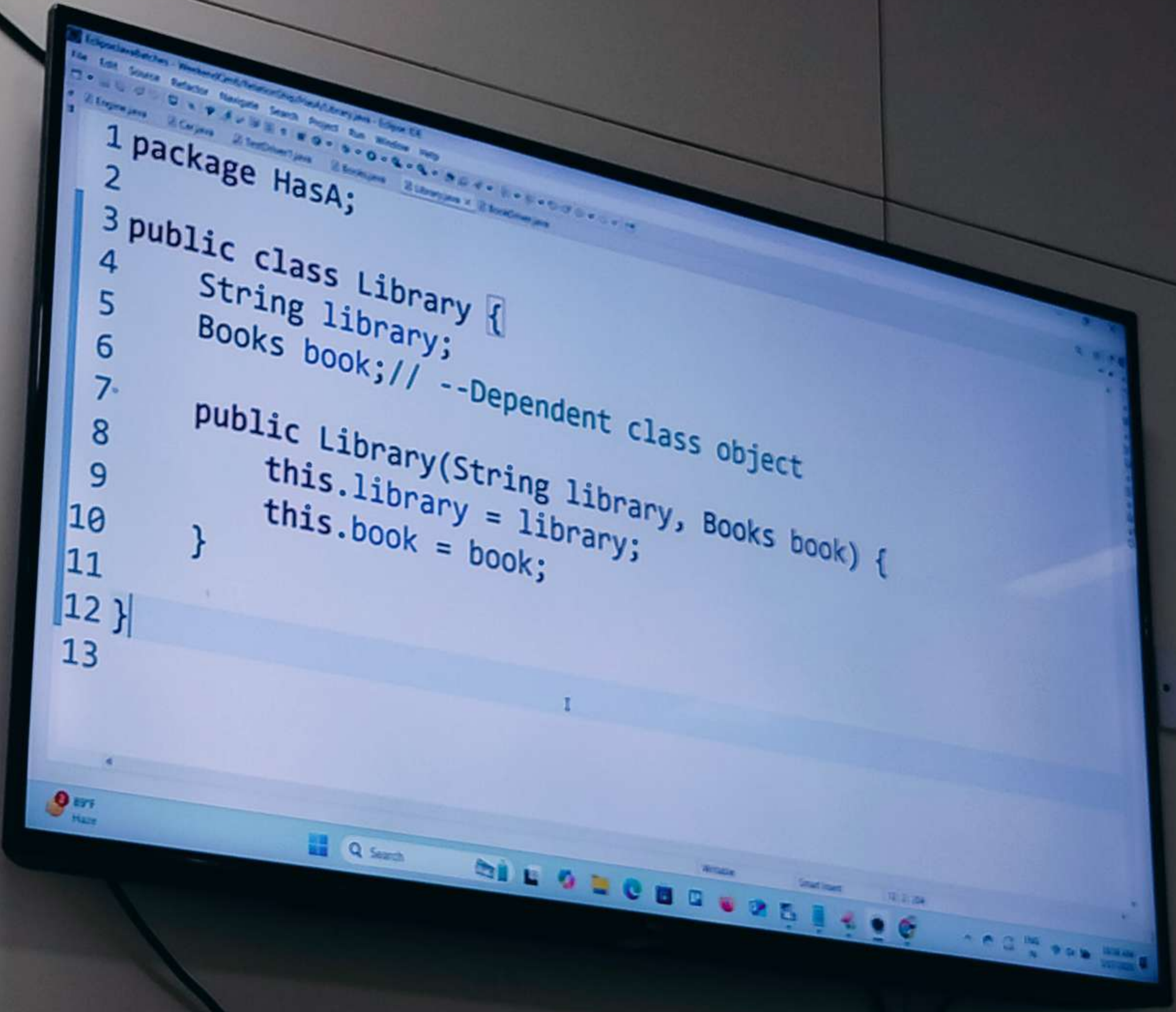
```
package HasA;

public class Car {
    String model="Punch";

    //Dependent class obj ref
    Engine e=new Engine();
}
```



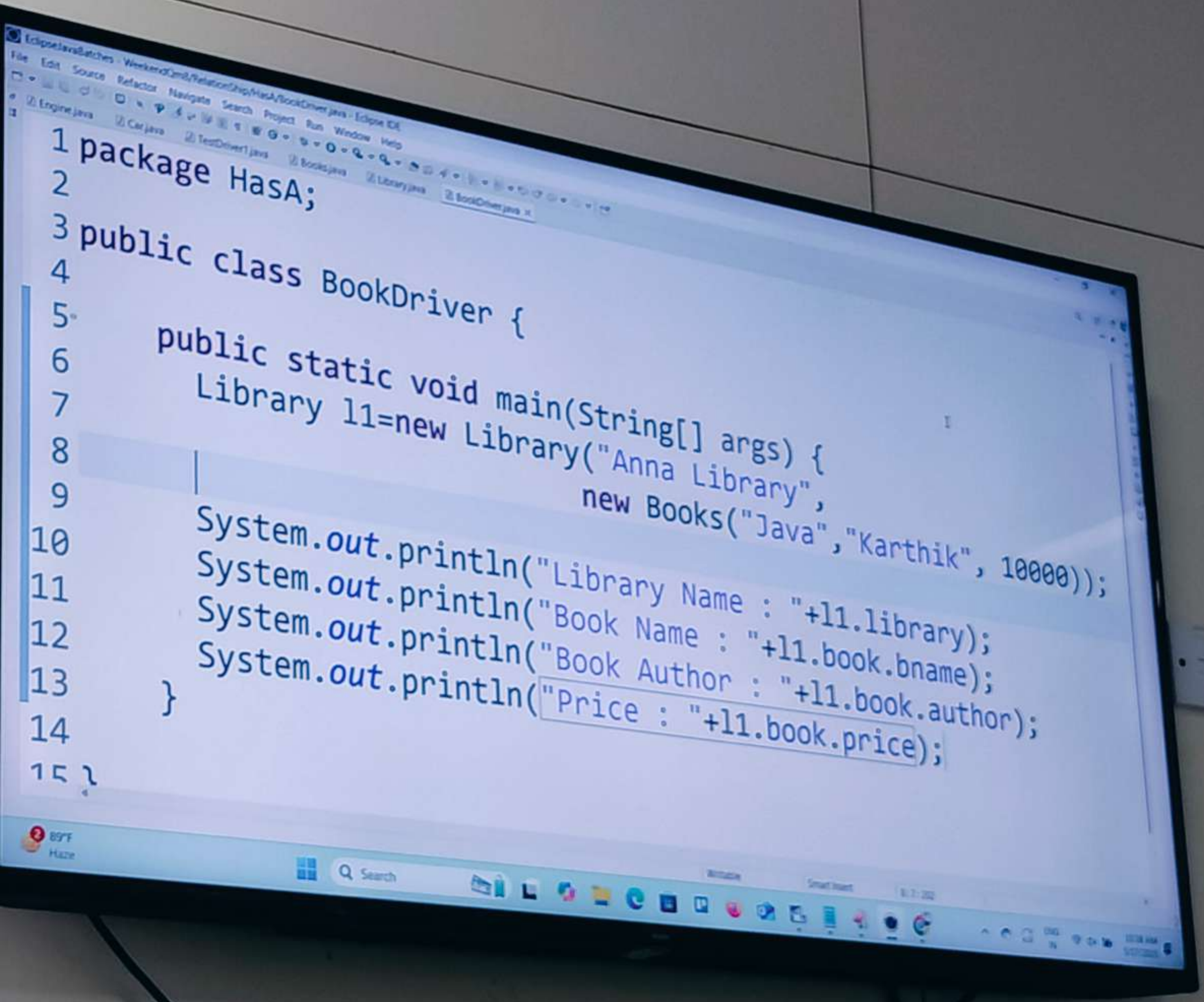
```java
package HasA;

public class TestDriver1 {

    public static void main(String[] args) {
        Car c1=new Car();
        System.out.println("Car Model : "+c1.model);
        System.out.println("Car Engine: "+c1.e.eno);

        c1=null;
        //-->NullPointerException
        System.out.println("Car Model : "+c1.model);
        System.out.println("Car Engine: "+c1.e.eno);
    }
```



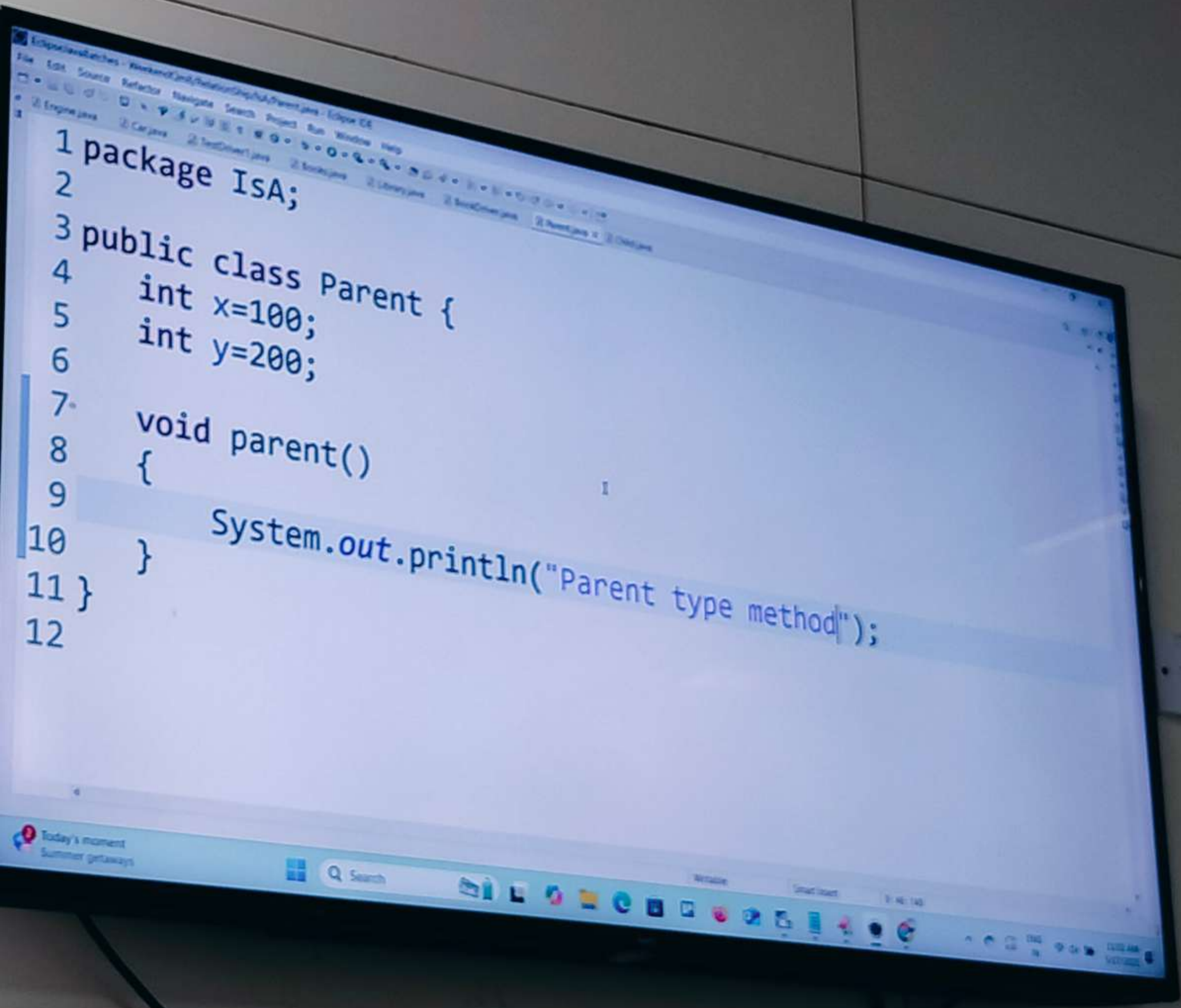
```java
package HasA;

public class Books {
    String bname, author;
    double price;

    public Books(String bname, String author, double price) {
        this.bname = bname;
        this.author = author;
        this.price = price;
    }
}

```



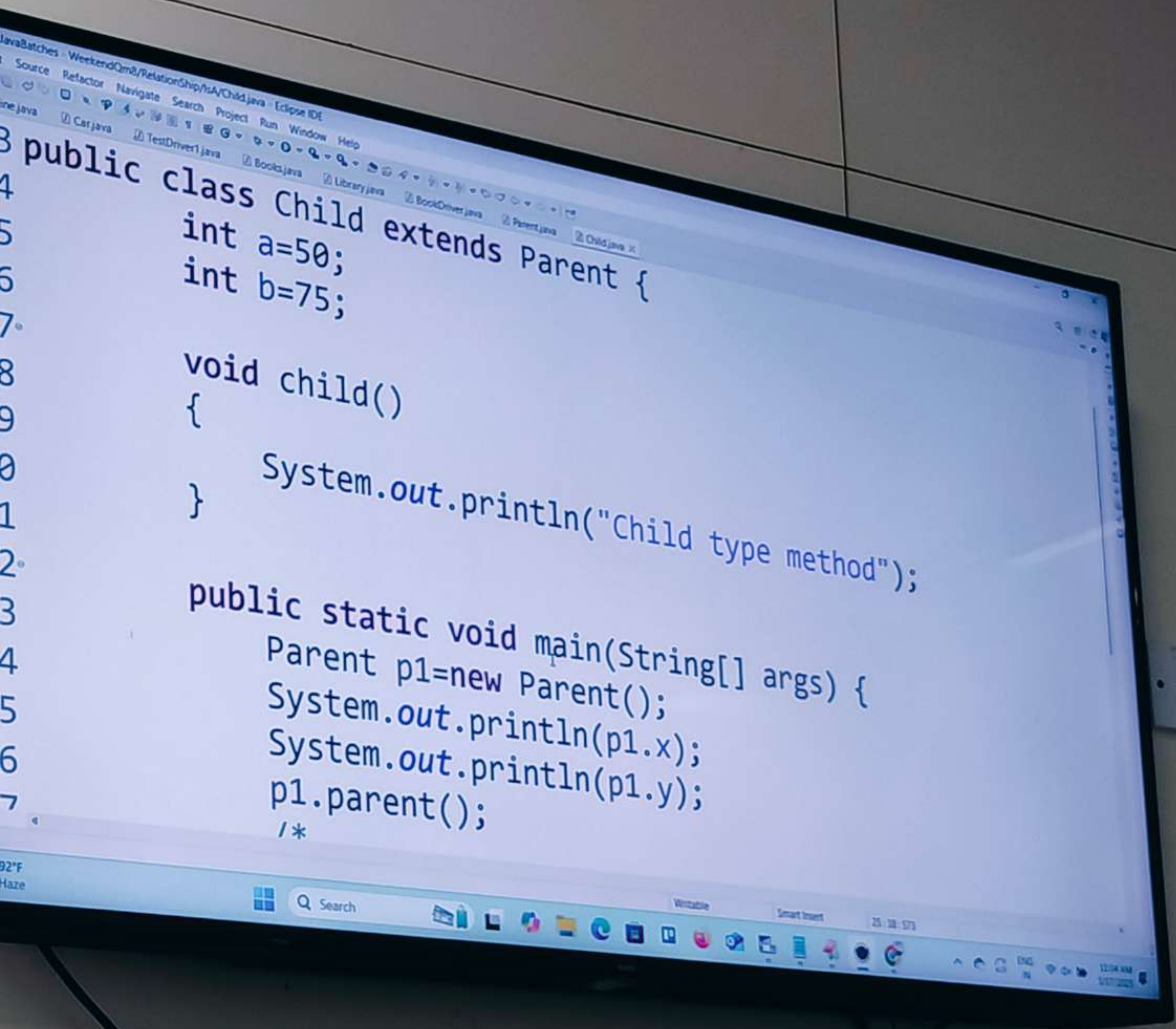
```java
package HasA;

public class Library {
    String library;
    Books book;// --Dependent class object

    public Library(String library, Books book) {
        this.library = library;
        this.book = book;
    }

}
```


EclipseJavaBatches - WeekendQm8/RelationShip/HasA/BookDriver.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Engine.java
Car.java
TestDriver1.java
Books.java
Library.java
BookDriver.java
89°F
Haze
Search
Writable
Smart Insert

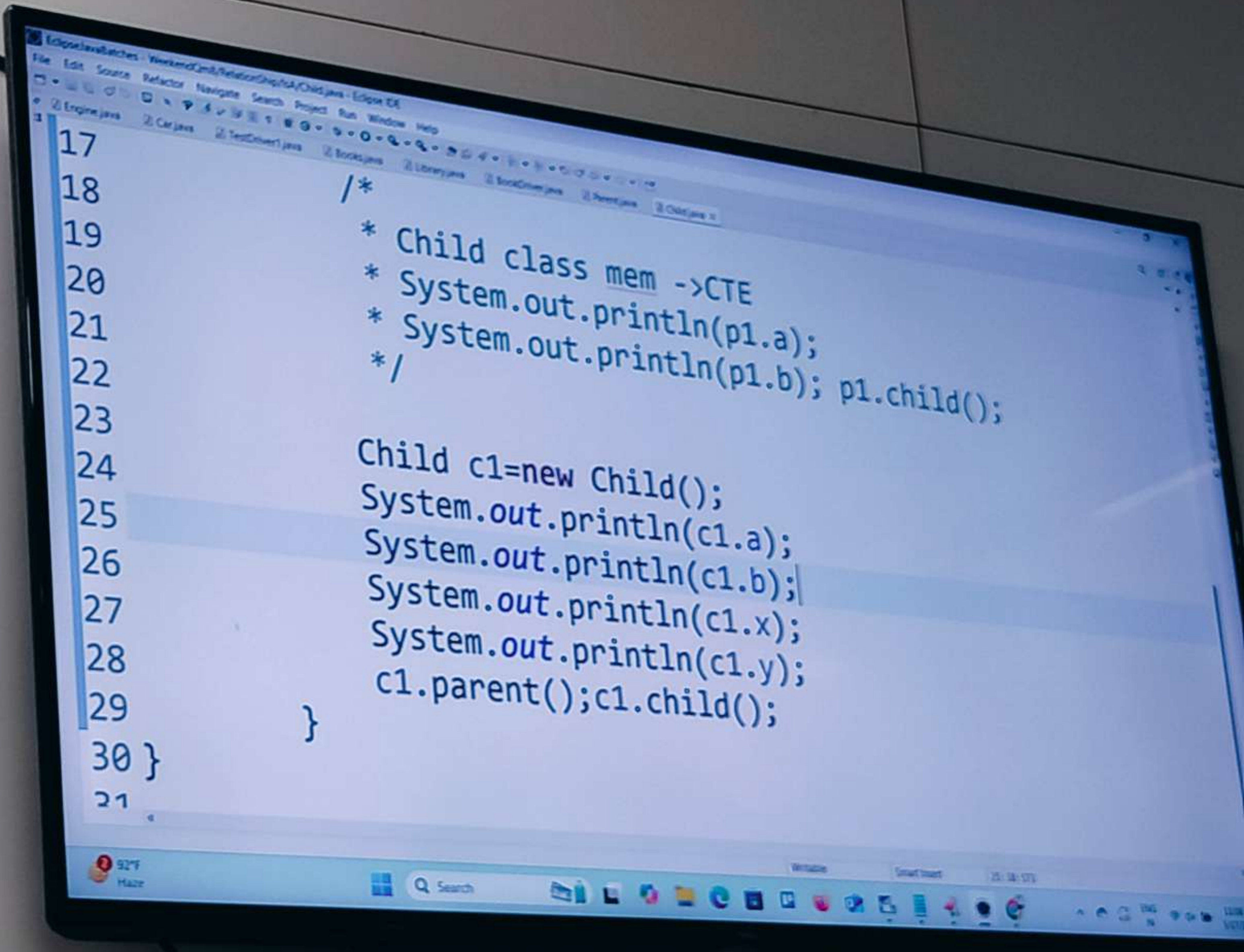
```java
package HasA;

public class BookDriver {

    public static void main(String[] args) {
        Library l1=new Library("Anna Library",

                              new Books("Java","Karthik", 10000));
        System.out.println("Library Name : "+l1.library);
        System.out.println("Book Name : "+l1.book.bname);
        System.out.println("Book Author : "+l1.book.author);
        System.out.println("Price : "+l1.book.price);
    }

}
```



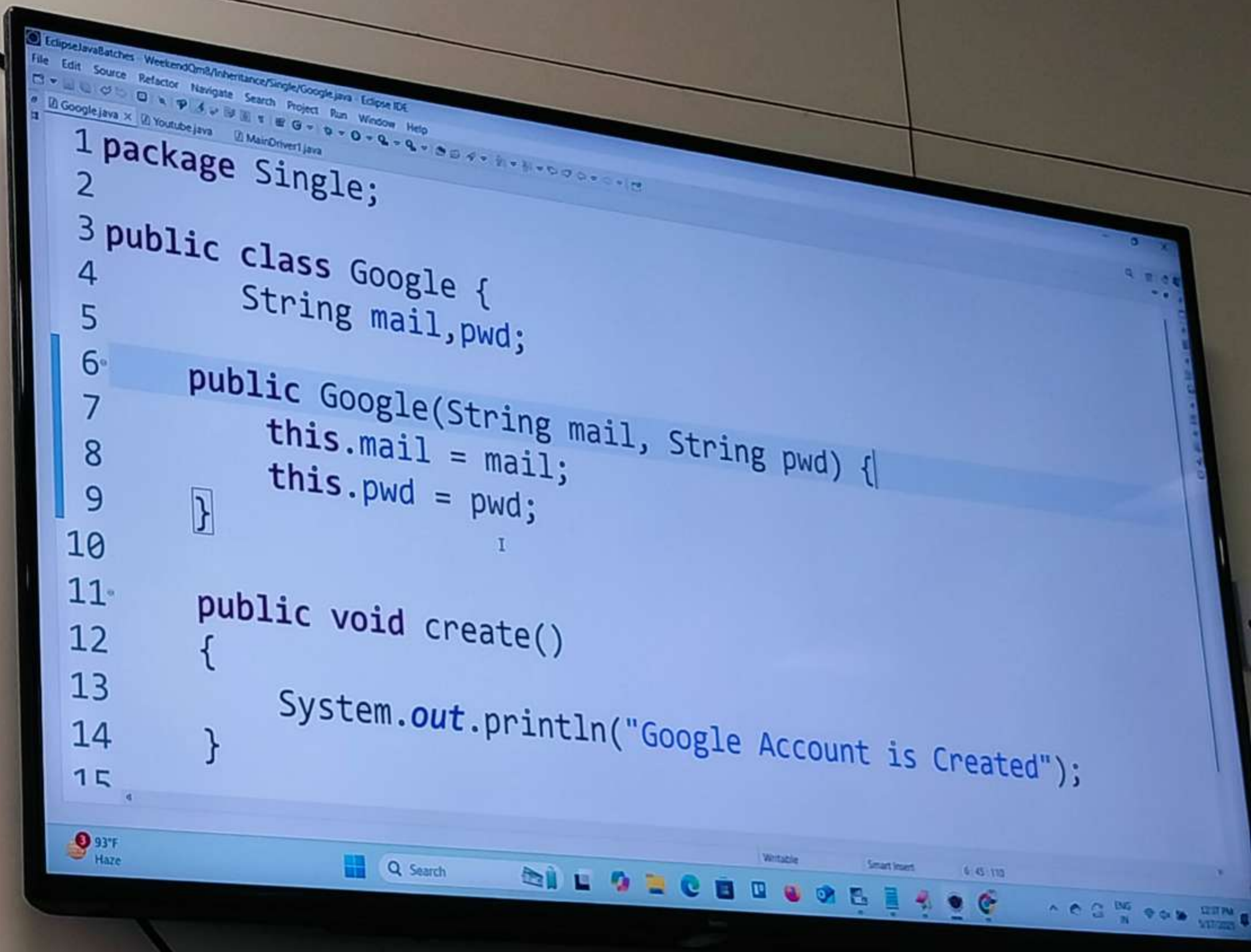
```java
package IsA;

public class Parent {
    int x=100;
    int y=200;

    void parent()
    {
        System.out.println("Parent type method");
    }
}

```



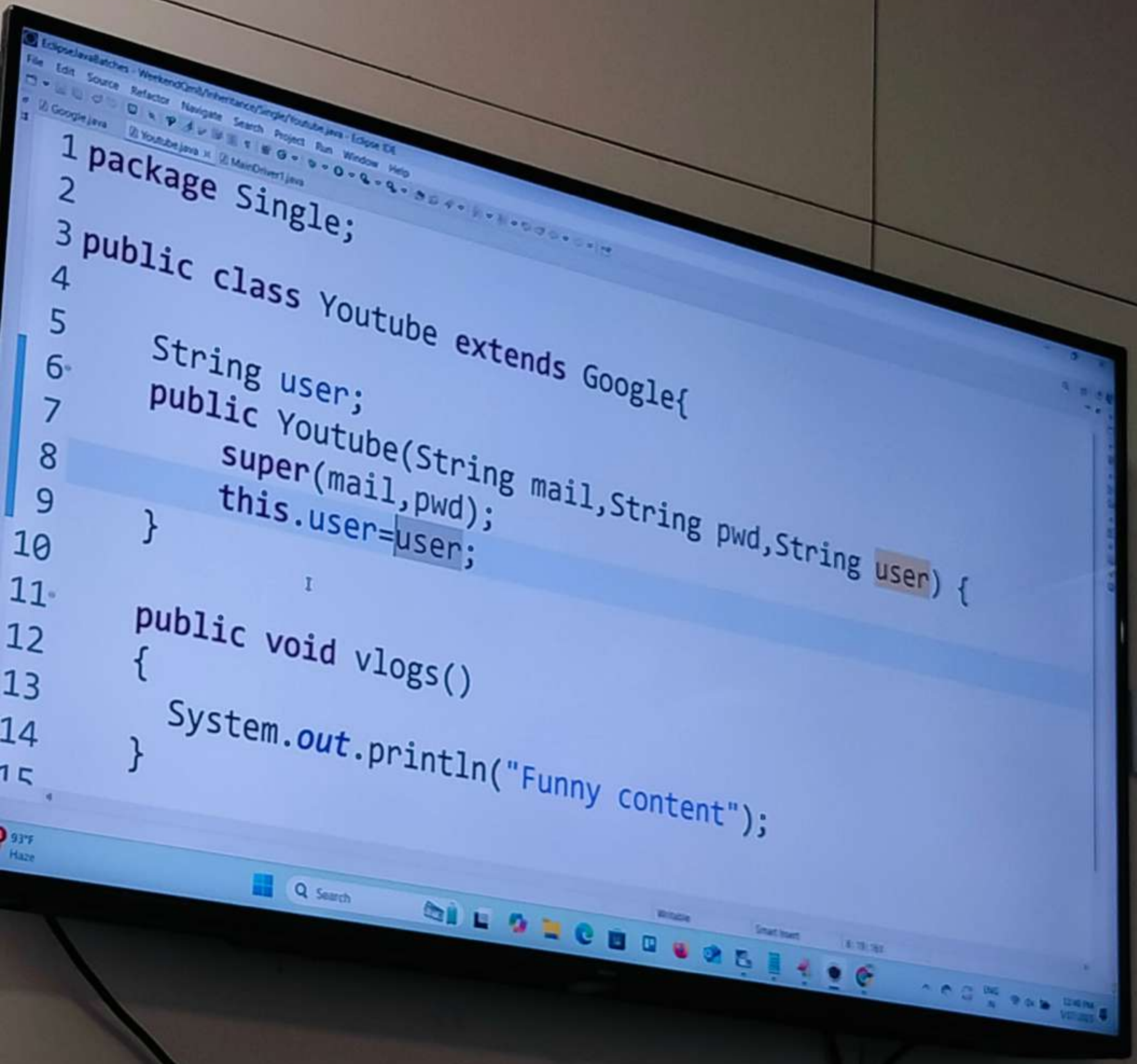
```
public class Child extends Parent {
        int a=50;
        int b=75;

        void child()
        {
            System.out.println("Child type method");
        }

        public static void main(String[] args) {
            Parent p1=new Parent();
            System.out.println(p1.x);
            System.out.println(p1.y);
            p1.parent();
            /*
```


EclipseJavaBatches - WeekendCjmB/RelationShip/IsA/Child.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Engine.java
Car.java
TestDriver1.java
Books.java
Library.java
BookDriver.java
Parent.java
Child.java
92°F
Haze
Search

```
                /*
                 * Child class mem ->CTE
                 * System.out.println(p1.a);
                 * System.out.println(p1.b); p1.child();
                 */

                Child c1=new Child();
                System.out.println(c1.a);
                System.out.println(c1.b);
                System.out.println(c1.x);
                System.out.println(c1.y);
                c1.parent();c1.child();
        }
}
```

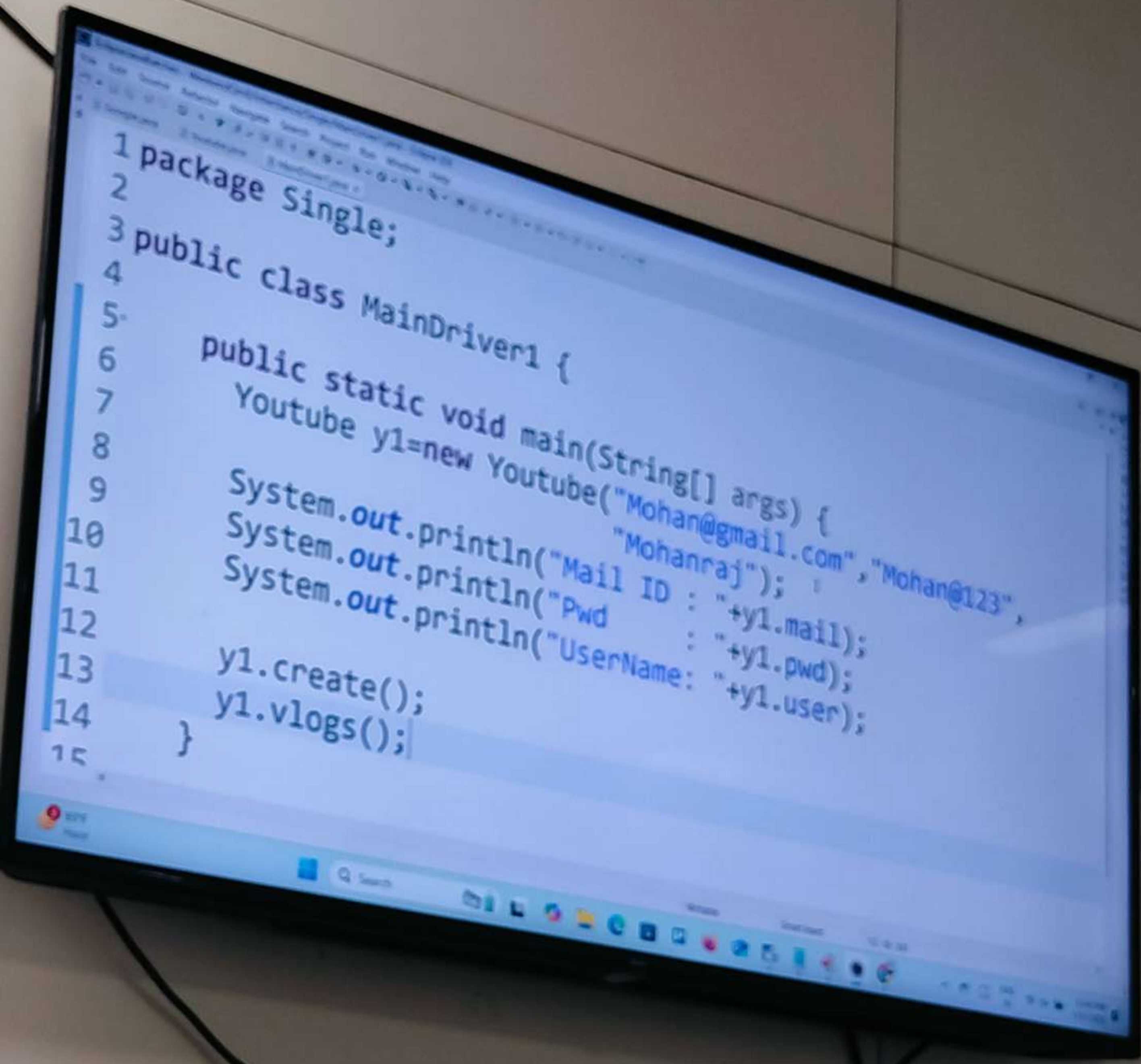

EclipseJavaBatches - WeekendQm8/Inheritance/Single/Google.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Google.java
Youtube.java
MainDriver1.java
Writable
Smart Insert
6 : 45 : 110
93°F
Haze
Search

```java
package Single;

public class Google {
        String mail,pwd;

    public Google(String mail, String pwd) {
        this.mail = mail;
        this.pwd = pwd;
    }

    public void create()
    {
        System.out.println("Google Account is Created");
    }
```

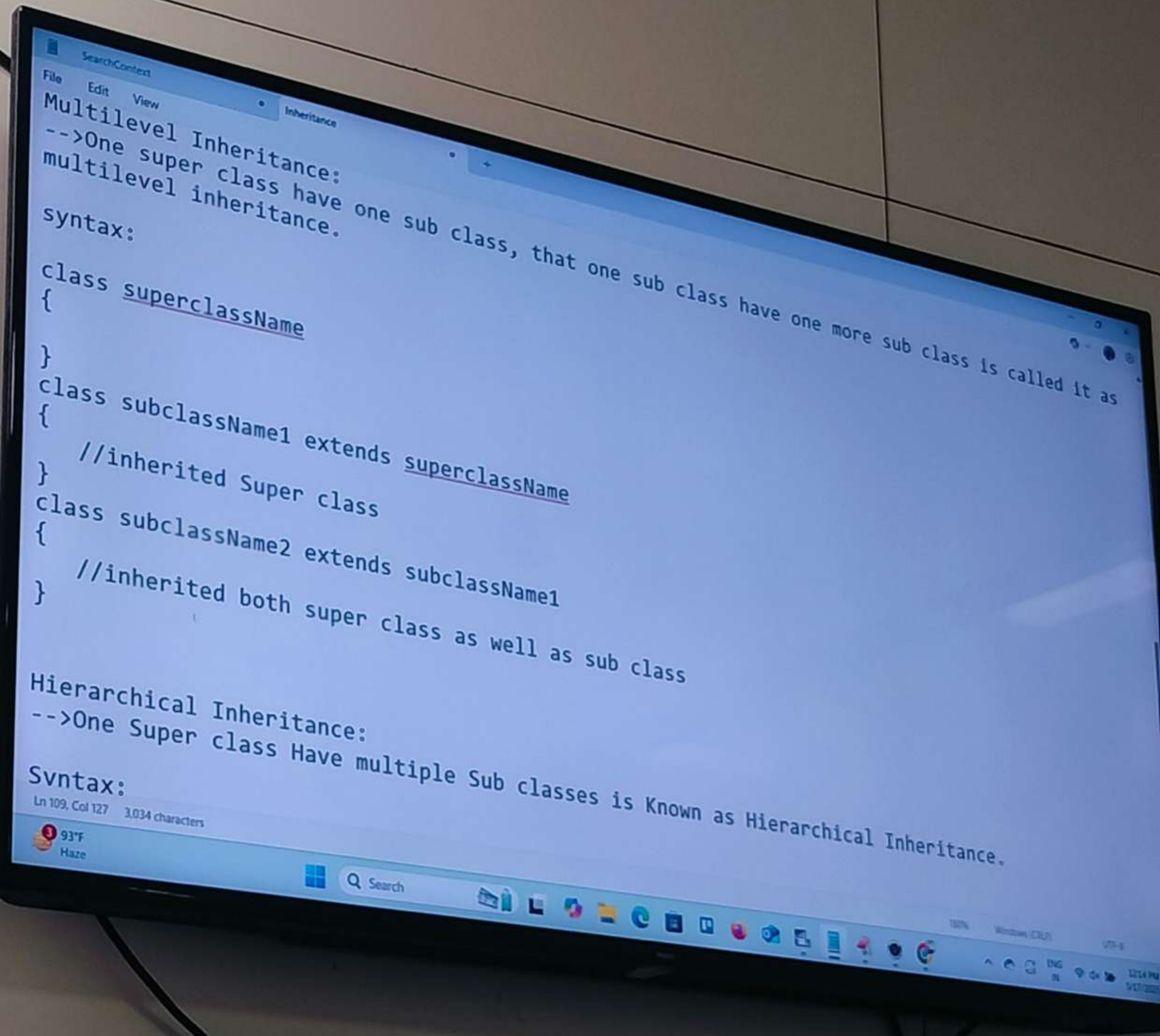


EclipseJavaBatches - WeekendQmB/Inheritance/Single/Youtube.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Google.java
Youtube.java
MainDriver1.java
93°F
Haze
Search
Writable
Smart Insert

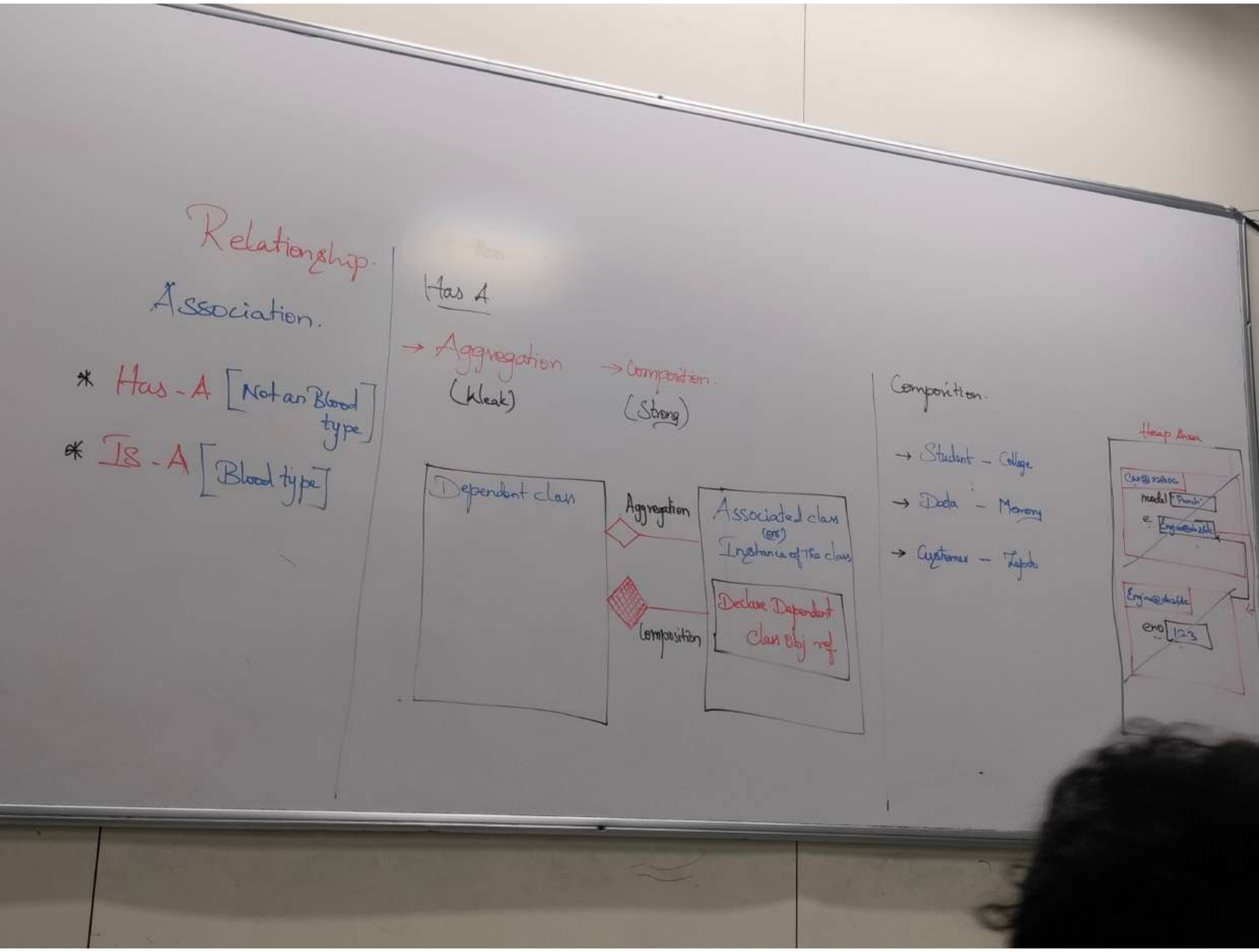
```java
package Single;
public class Youtube extends Google{

    String user;
    public Youtube(String mail,String pwd,String user) {
        super(mail,pwd);
        this.user=user;
    }

    public void vlogs()
    {
     System.out.println("Funny content");
    }
```



```
package Single;
public class MainDriver1 {
    public static void main(String[] args) {
        Youtube y1=new Youtube("Mohan@gmail.com","Mohan@123",
                "Mohanraj");
        System.out.println("Mail ID : "+y1.mail);
        System.out.println("Pwd      : "+y1.pwd);
        System.out.println("UserName: "+y1.user);
        y1.create();
        y1.vlogs();
    }
```



```
Multilevel Inheritance:
-->One super class have one sub class, that one sub class have one more sub class is called it as multilevel inheritance.
syntax:

class superclassName
{

}
class subclassName1 extends superclassName
{
   //inherited Super class
}
class subclassName2 extends subclassName1
{
   //inherited both super class as well as sub class
}


Hierarchical Inheritance:
-->One Super class Have multiple Sub classes is Known as Hierarchical Inheritance.

Syntax:
```



Relationship.

Association.

* Has-A [Not an Blood type]

* IS-A [Blood type]

Has A

→ Aggregation (Weak)

→ Composition. (Strong)

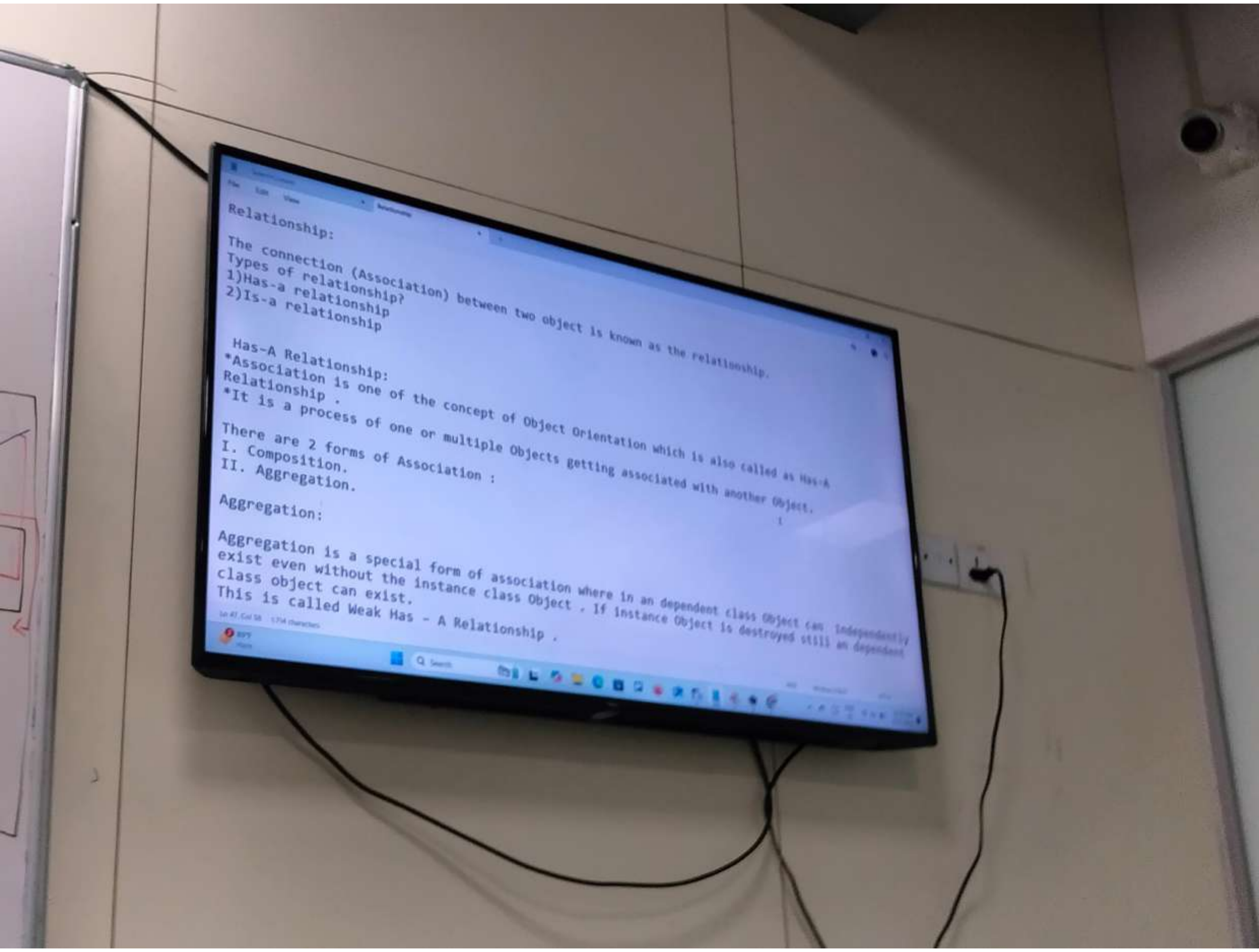
Dependent class
Aggregation
Composition
Associated class (or) Instance of The class
Declare Dependent class Obj ref

Composition.

→ Student – College

→ Data – Memory

→ Customer – Zepto



Relationship:

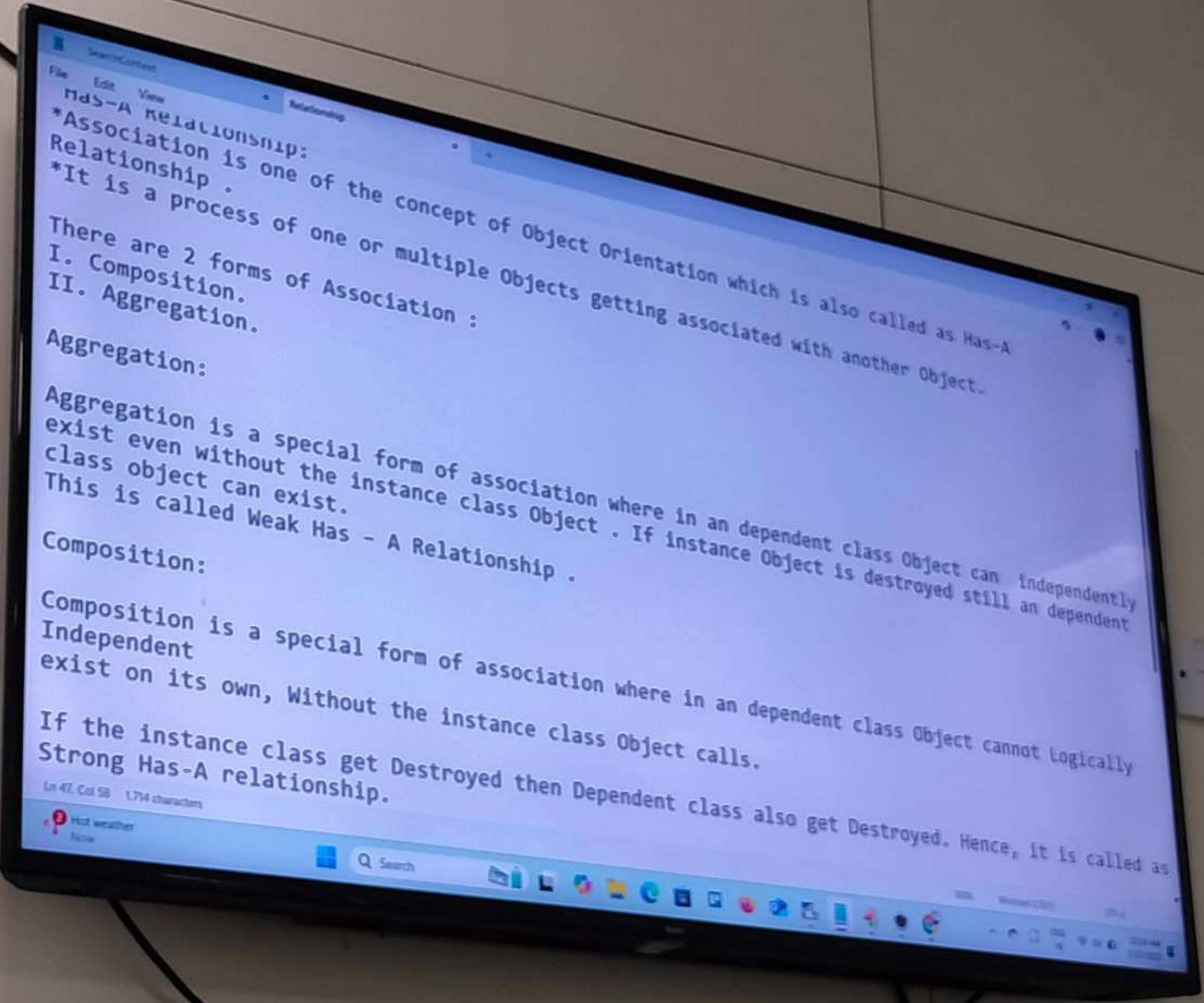
The connection (Association) between two object is known as the relationship.
Types of relationship?
1)Has-a relationship
2)Is-a relationship

Has-A Relationship:
*Association is one of the concept of Object Orientation which is also called as Has-A Relationship .
*It is a process of one or multiple Objects getting associated with another Object.

There are 2 forms of Association :
I. Composition.
II. Aggregation.

Aggregation:

Aggregation is a special form of association where in an dependent class Object can independently exist even without the instance class Object . If instance Object is destroyed still an dependent class object can exist.
This is called Weak Has - A Relationship .



Has-A Relationship:

*Association is one of the concept of Object Orientation which is also called as Has-A Relationship .

*It is a process of one or multiple Objects getting associated with another Object.

There are 2 forms of Association :

I. Composition.

II. Aggregation.

Aggregation:

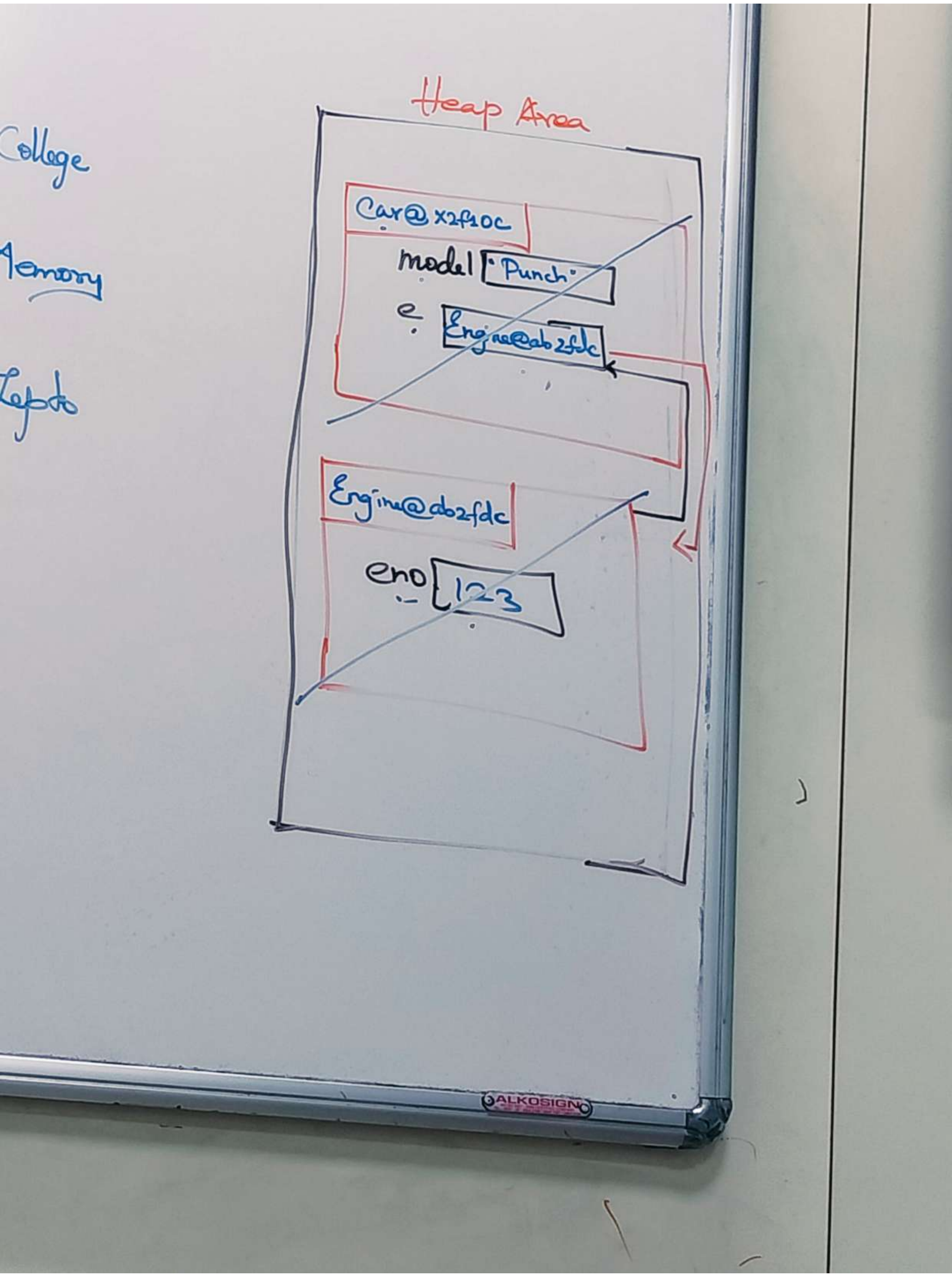
Aggregation is a special form of association where in an dependent class Object can independently exist even without the instance class Object . If instance Object is destroyed still an dependent class object can exist.

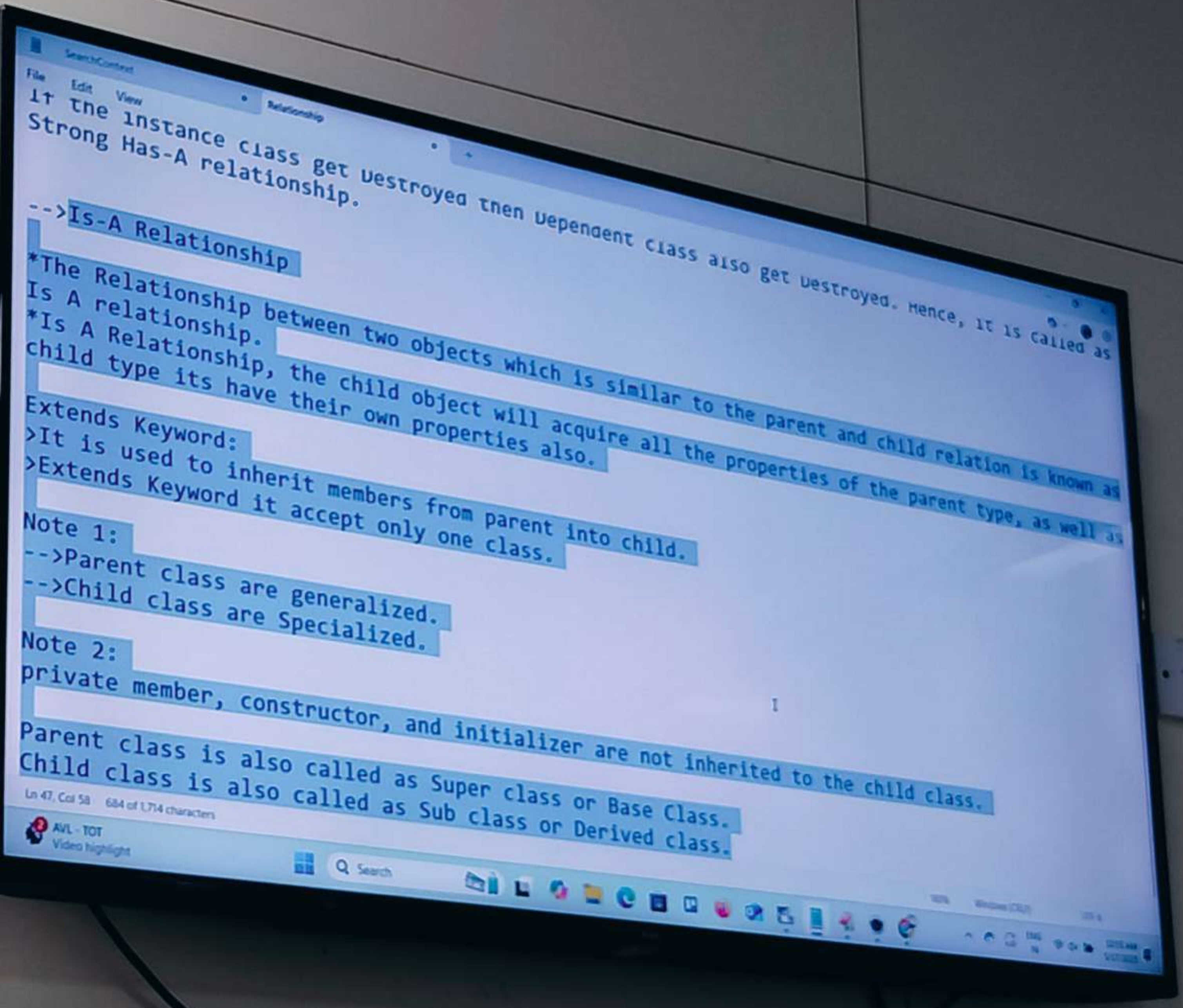
This is called Weak Has - A Relationship .

Composition:

Composition is a special form of association where in an dependent class Object cannot Logically Independent exist on its own, Without the instance class Object calls.

If the instance class get Destroyed then Dependent class also get Destroyed. Hence, it is called as Strong Has-A relationship.


College
Memory
Zepto
Heap Area
Car@x2f10c
model
"Punch"
e
Engine@ab2fdc
Engine@ab2fdc
eno
123
ALKOSIGN



If the instance class get Destroyed then Dependent class also get Destroyed. Hence, it is called as
Strong Has-A relationship.

-->Is-A Relationship

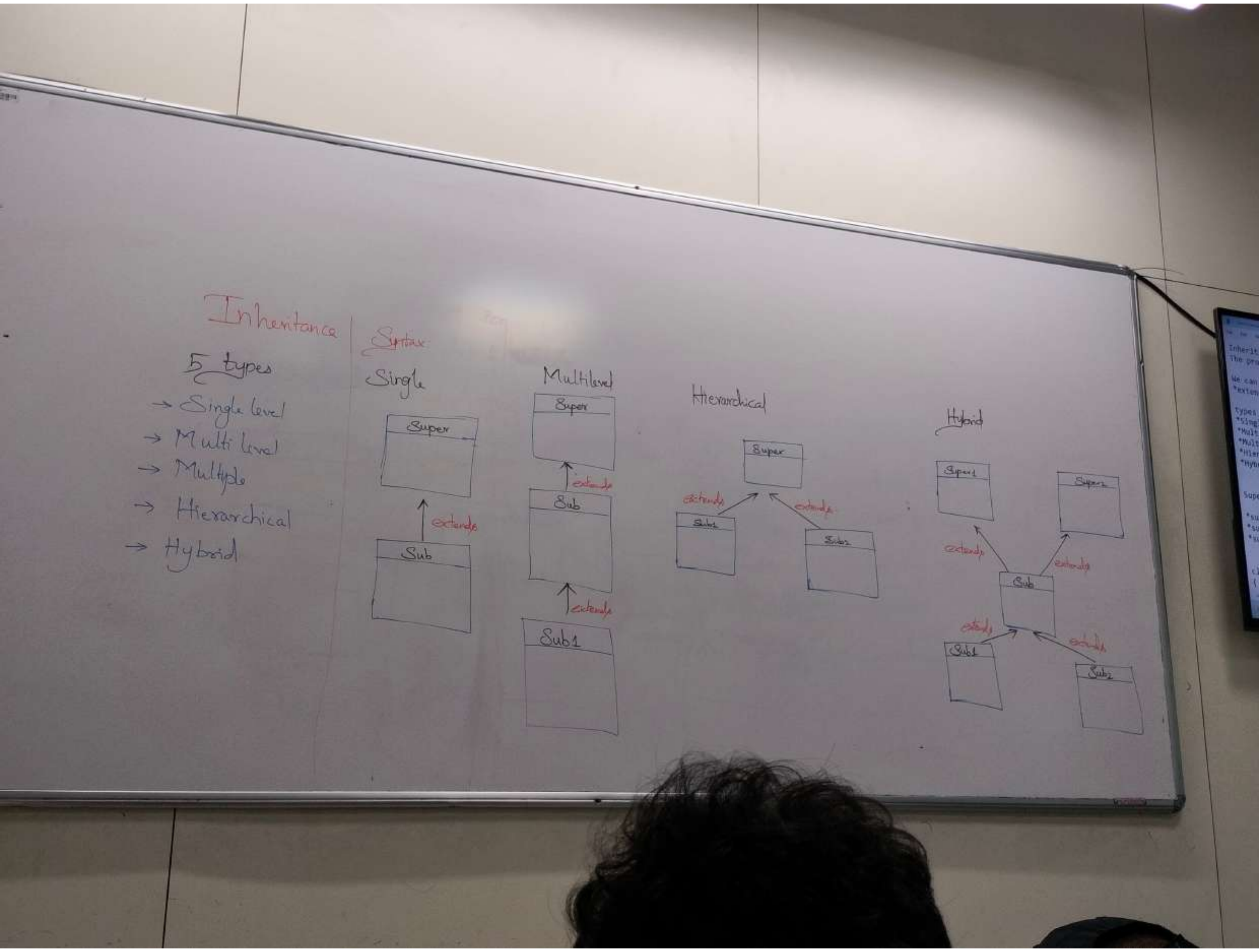
*The Relationship between two objects which is similar to the parent and child relation is known as
Is A relationship.
*Is A Relationship, the child object will acquire all the properties of the parent type, as well as
child type its have their own properties also.

Extends Keyword:
>It is used to inherit members from parent into child.
>Extends Keyword it accept only one class.

Note 1:
-->Parent class are generalized.
-->Child class are Specialized.

Note 2:
private member, constructor, and initializer are not inherited to the child class.

Parent class is also called as Super class or Base Class.
Child class is also called as Sub class or Derived class.

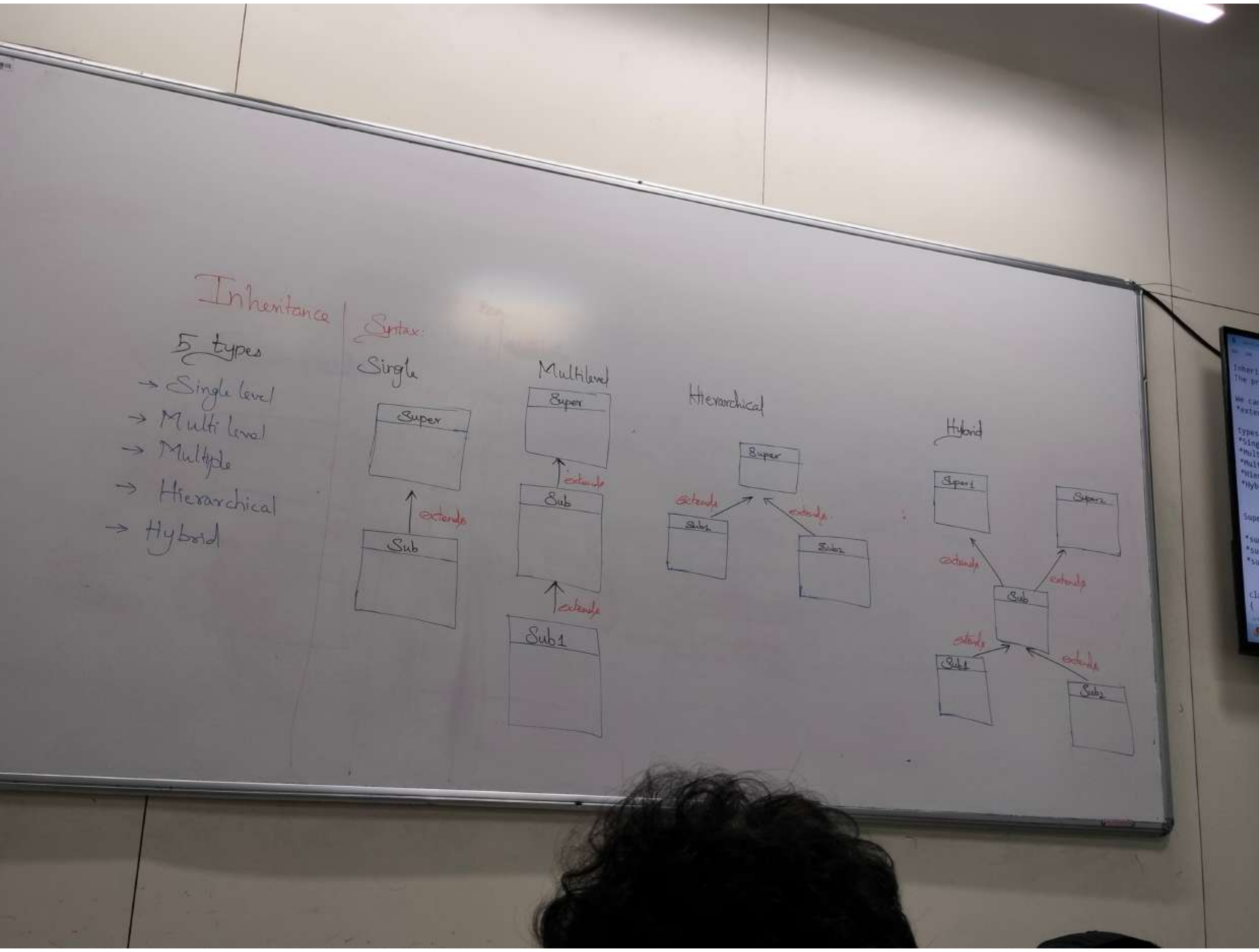


Inheritance

5 types

- → Single level
- → Multi level
- → Multiple
- → Hierarchical
- → Hybrid

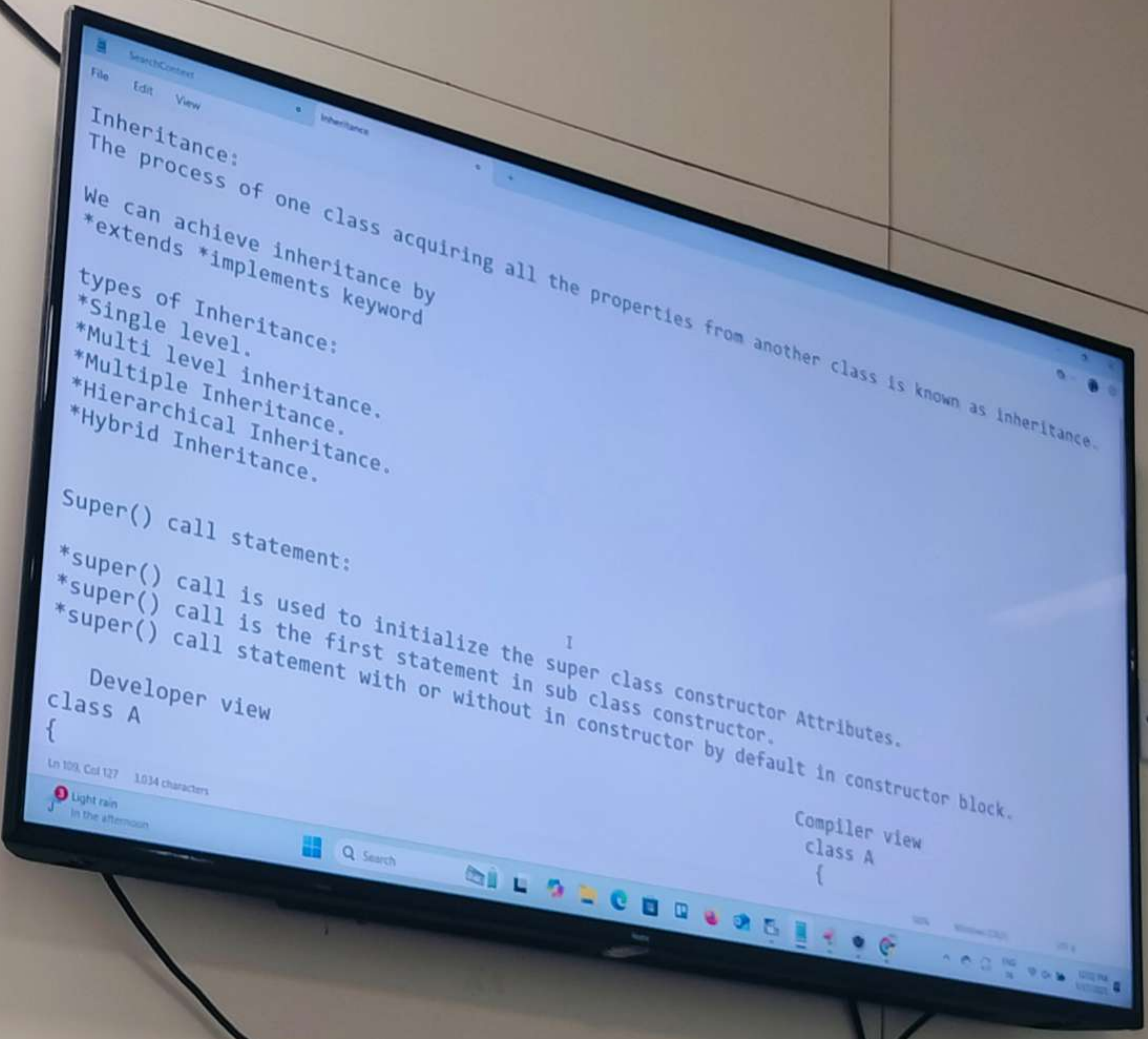
Syntax

# Inheritance

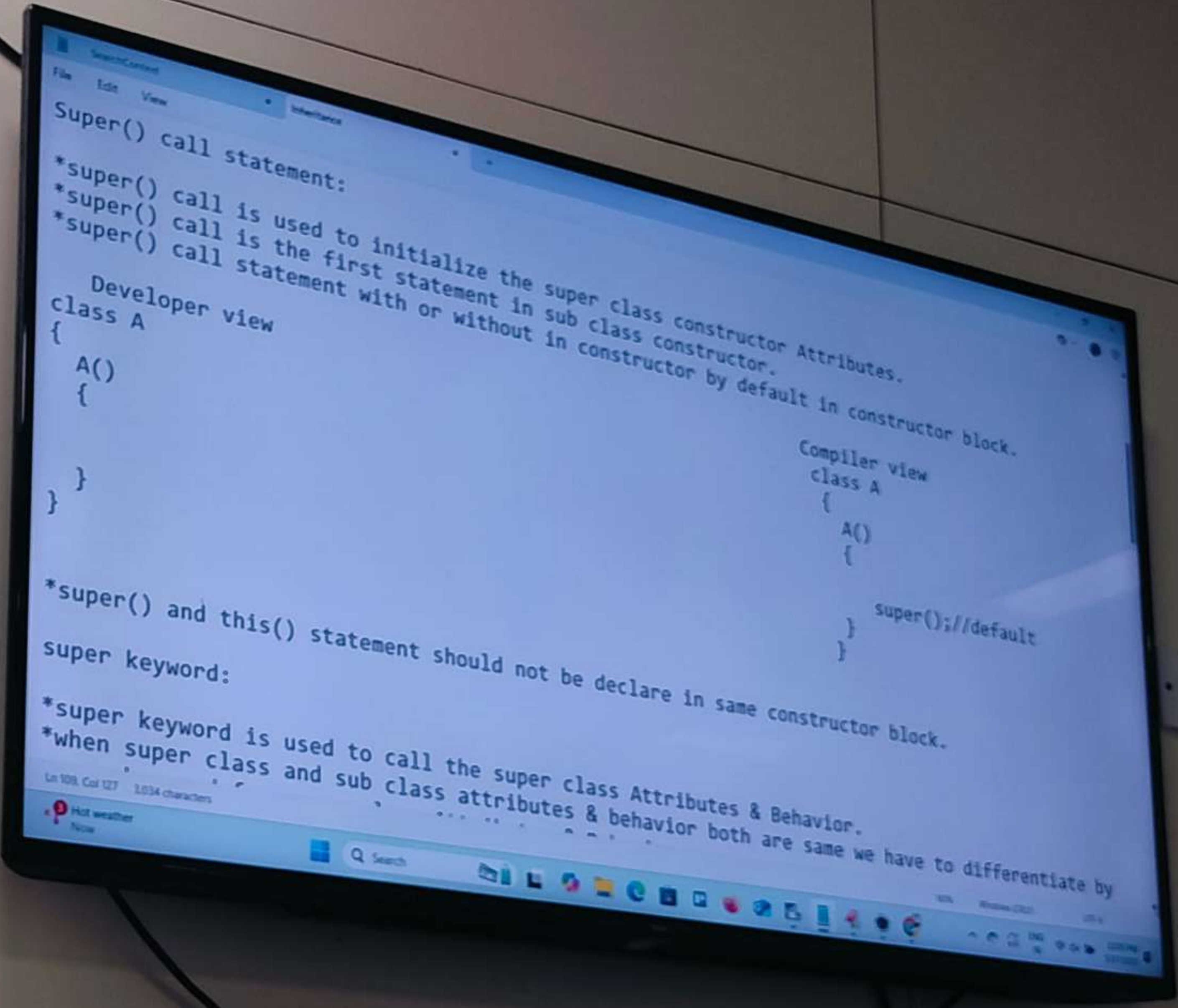
5 types

- → Single level
- → Multi level
- → Multiple
- → Hierarchical
- → Hybrid

Syntax:



```
Inheritance:
The process of one class acquiring all the properties from another class is known as inheritance.

We can achieve inheritance by
*extends *implements keyword

types of Inheritance:
*Single level.
*Multi level inheritance.
*Multiple Inheritance.
*Hierarchical Inheritance.
*Hybrid Inheritance.


Super() call statement:

*super() call is used to initialize the super class constructor Attributes.
*super() call is the first statement in sub class constructor.
*super() call statement with or without in constructor by default in constructor block.

   Developer view                                          Compiler view
class A                                                     class A
{                                                            {
```



Super() call statement:

*super() call is used to initialize the super class constructor Attributes.
*super() call is the first statement in sub class constructor.
*super() call statement with or without in constructor by default in constructor block.

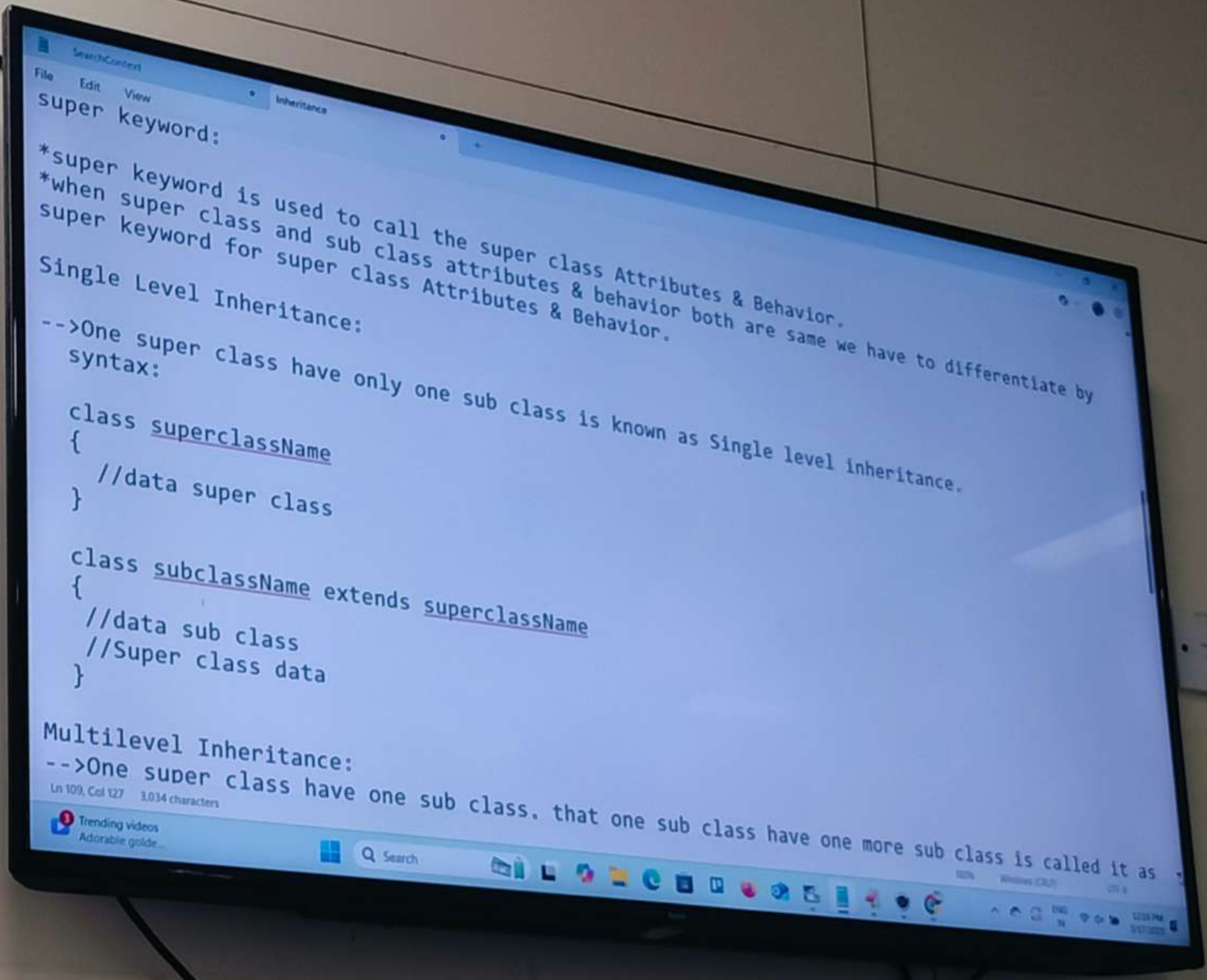
```
   Developer view
class A
{
  A()
  {


  }
}
```

```
Compiler view
 class A
 {
   A()
   {

     super();//default
   }
 }
```

*super() and this() statement should not be declare in same constructor block.

super keyword:

*super keyword is used to call the super class Attributes & Behavior.
*when super class and sub class attributes & behavior both are same we have to differentiate by



super keyword:

*super keyword is used to call the super class Attributes & Behavior.
*when super class and sub class attributes & behavior both are same we have to differentiate by
super keyword for super class Attributes & Behavior.

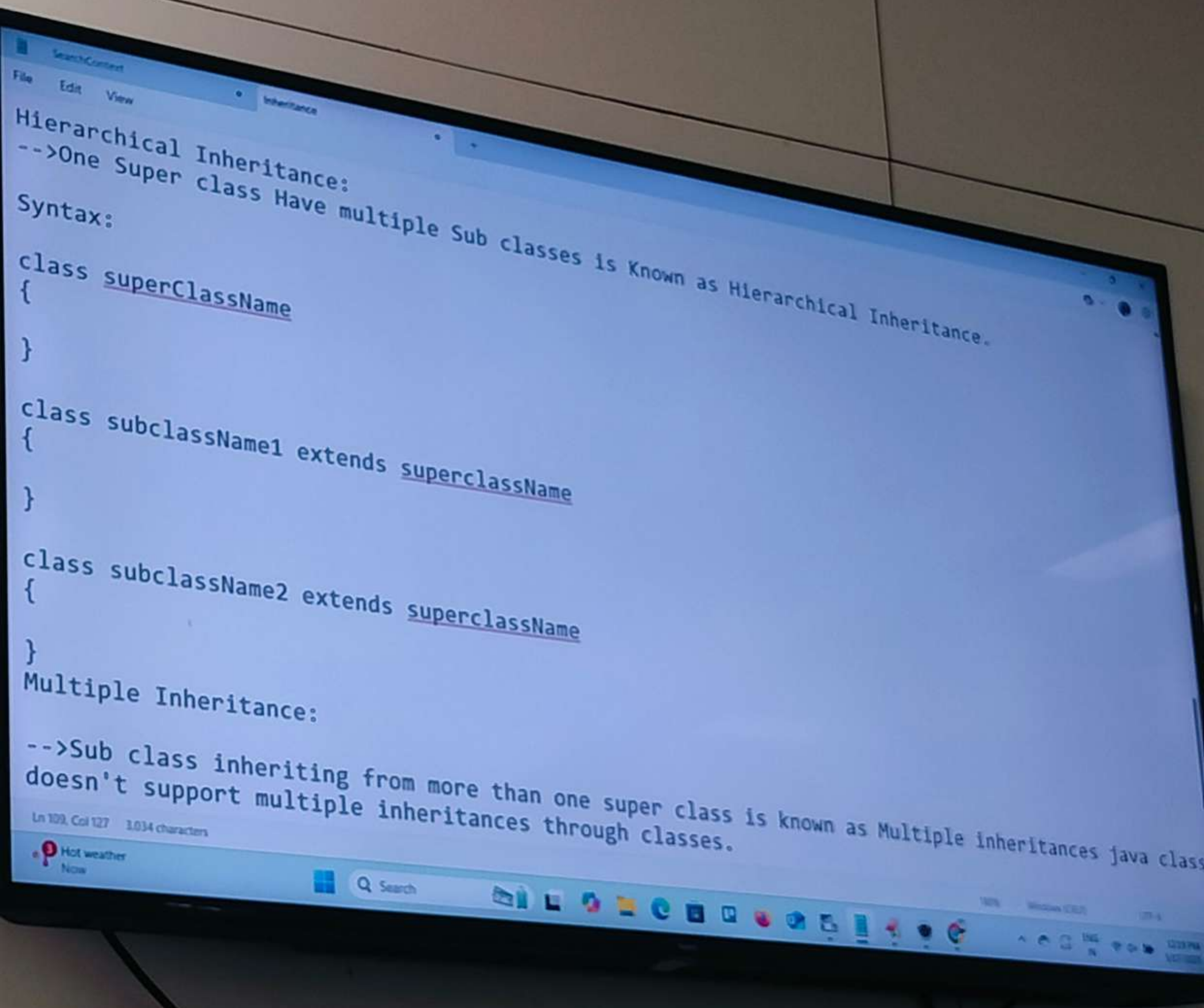
Single Level Inheritance:

-->One super class have only one sub class is known as Single level inheritance.
syntax:

```
  class superclassName
  {
    //data super class
  }

  class subclassName extends superclassName
  {
   //data sub class
   //Super class data
  }
```

Multilevel Inheritance:
-->One super class have one sub class. that one sub class have one more sub class is called it as

Ln 109, Col 127   3,034 characters

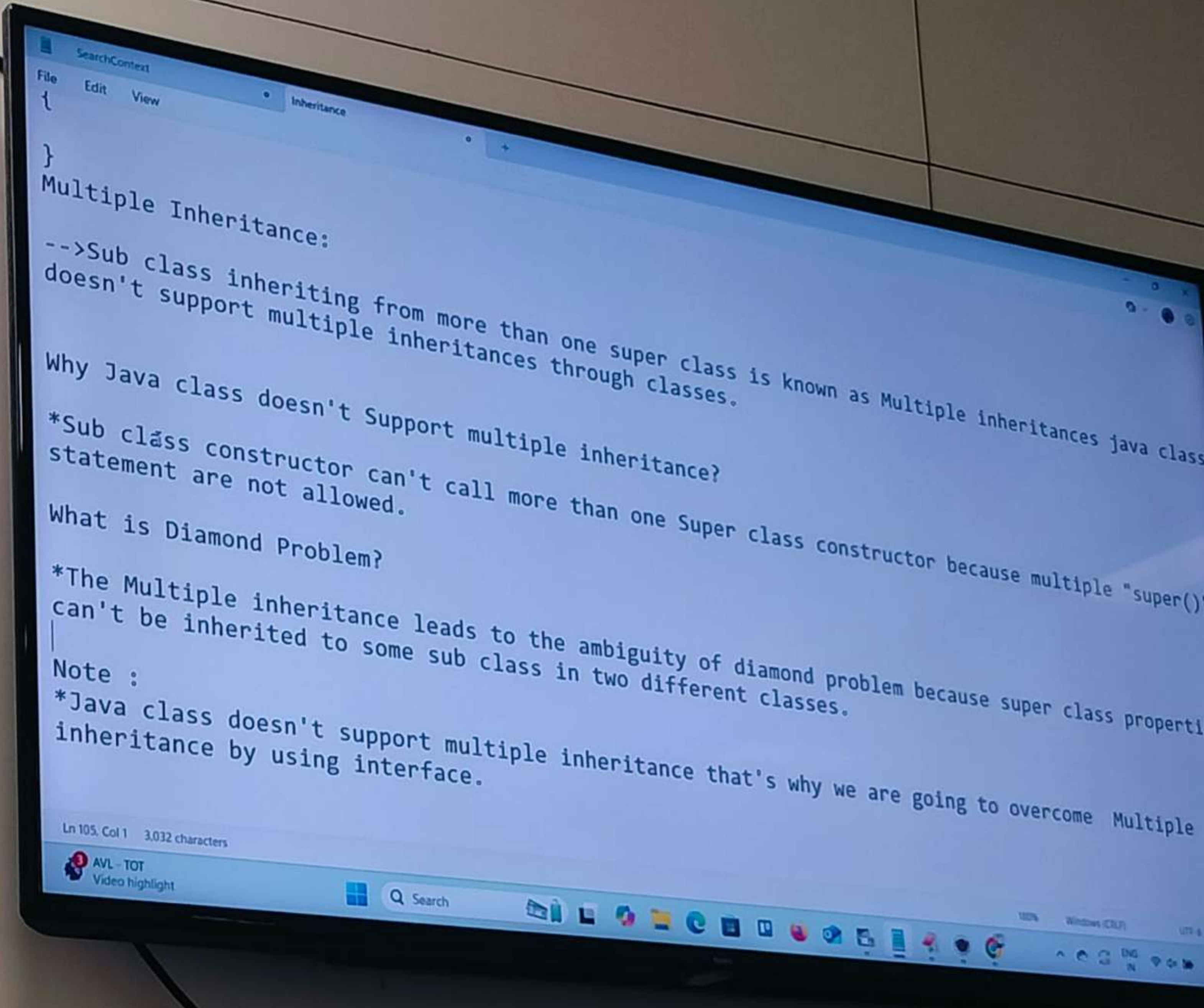


Hierarchical Inheritance:
-->One Super class Have multiple Sub classes is Known as Hierarchical Inheritance.
Syntax:

```
class superClassName
{

}

class subclassName1 extends superclassName
{

}

class subclassName2 extends superclassName
{

}
```

Multiple Inheritance:

-->Sub class inheriting from more than one super class is known as Multiple inheritances java class
doesn't support multiple inheritances through classes.



{

}

Multiple Inheritance:

-->Sub class inheriting from more than one super class is known as Multiple inheritances java class
doesn't support multiple inheritances through classes.

Why Java class doesn't Support multiple inheritance?

*Sub class constructor can't call more than one Super class constructor because multiple "super()
statement are not allowed.

What is Diamond Problem?

*The Multiple inheritance leads to the ambiguity of diamond problem because super class properti
can't be inherited to some sub class in two different classes.

Note :

*Java class doesn't support multiple inheritance that's why we are going to overcome Multiple
inheritance by using interface.