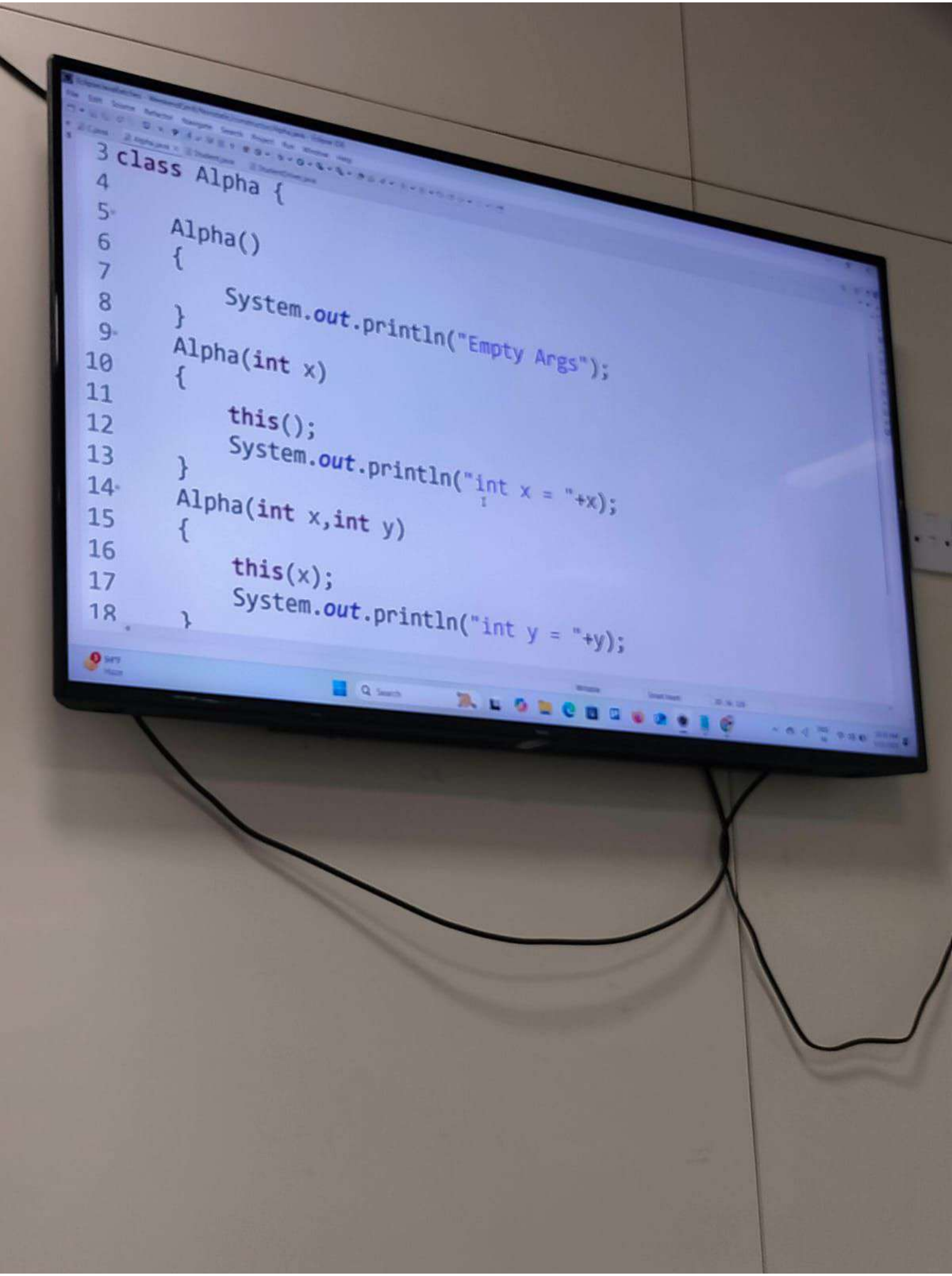
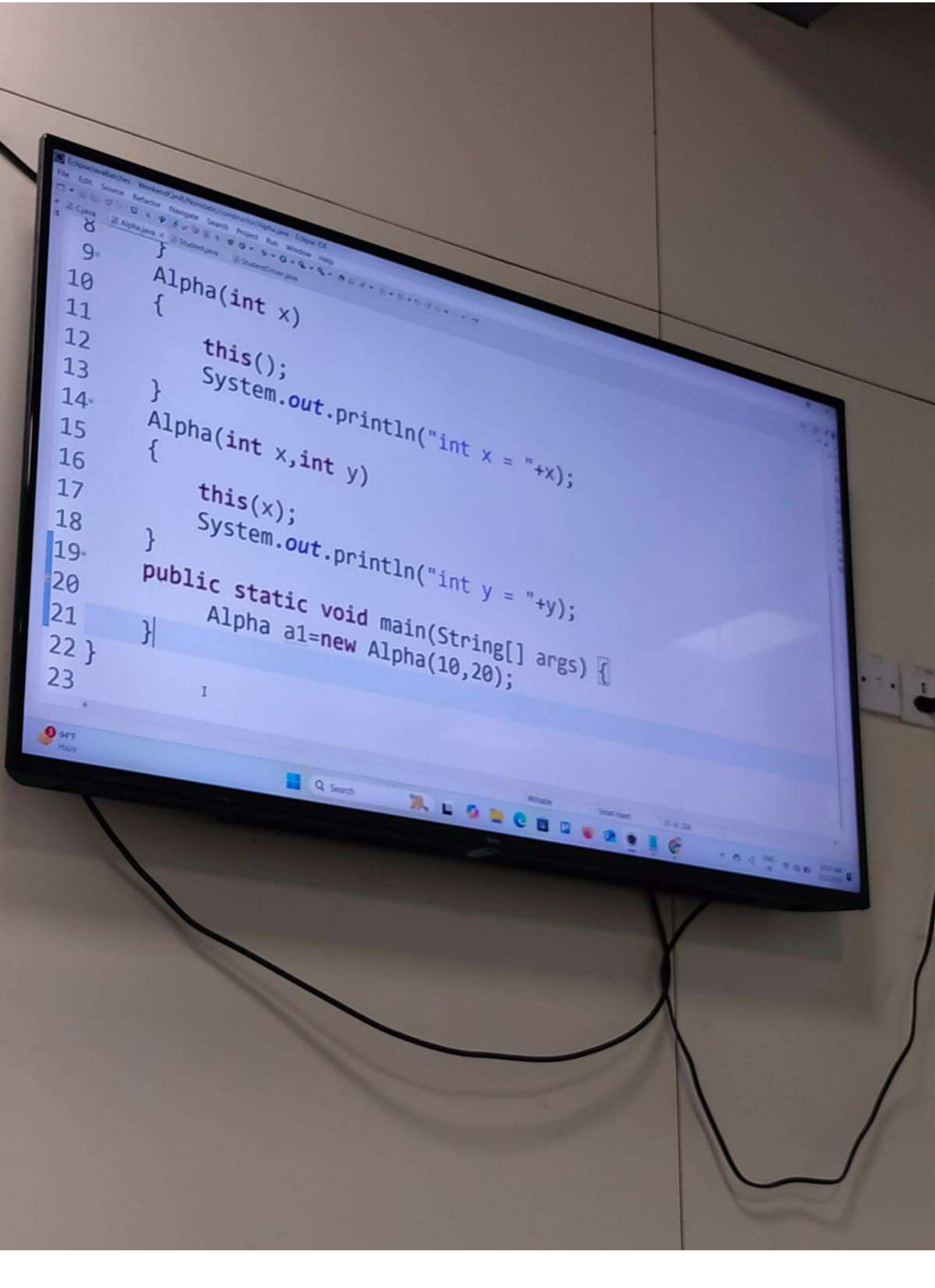
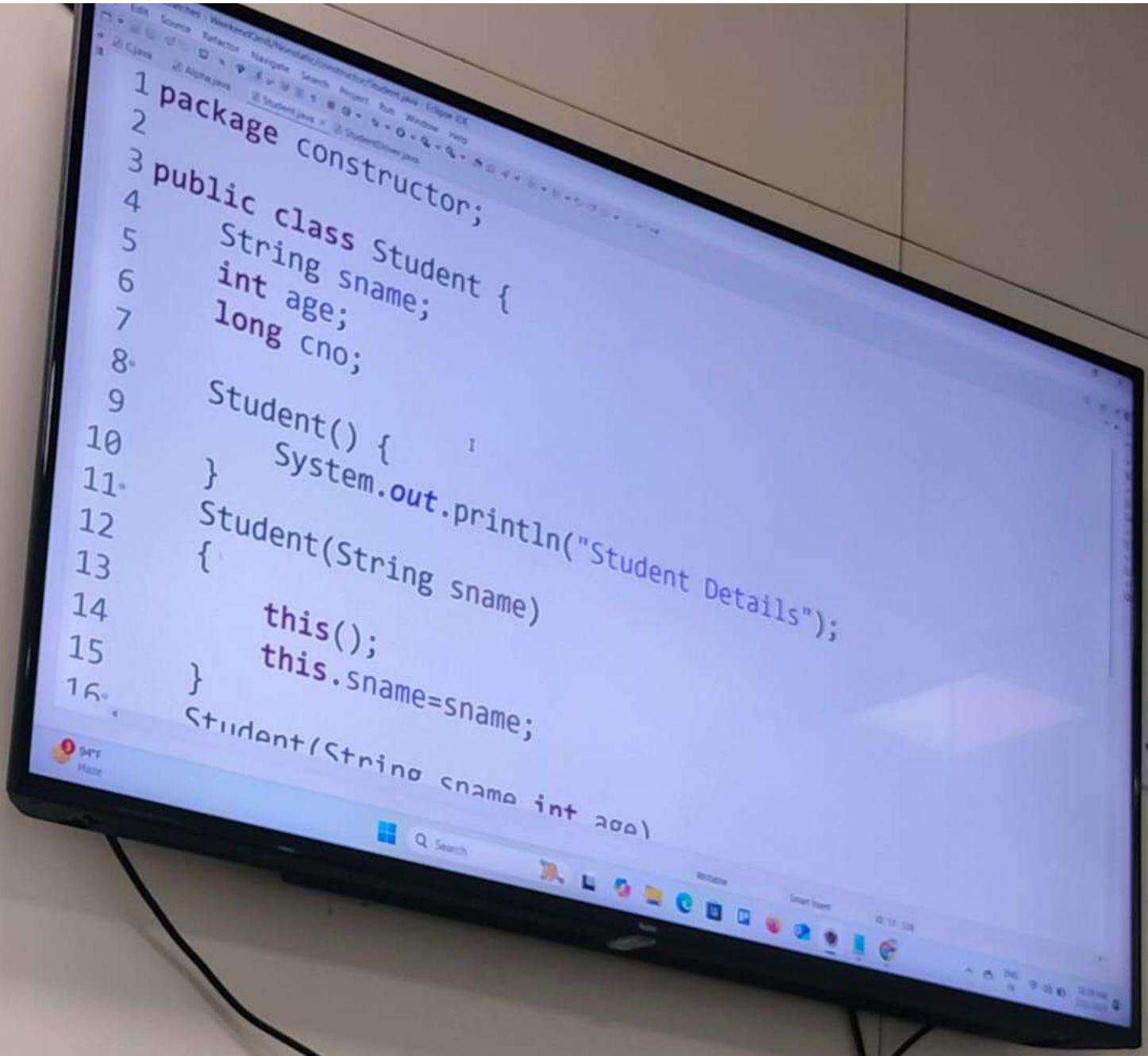
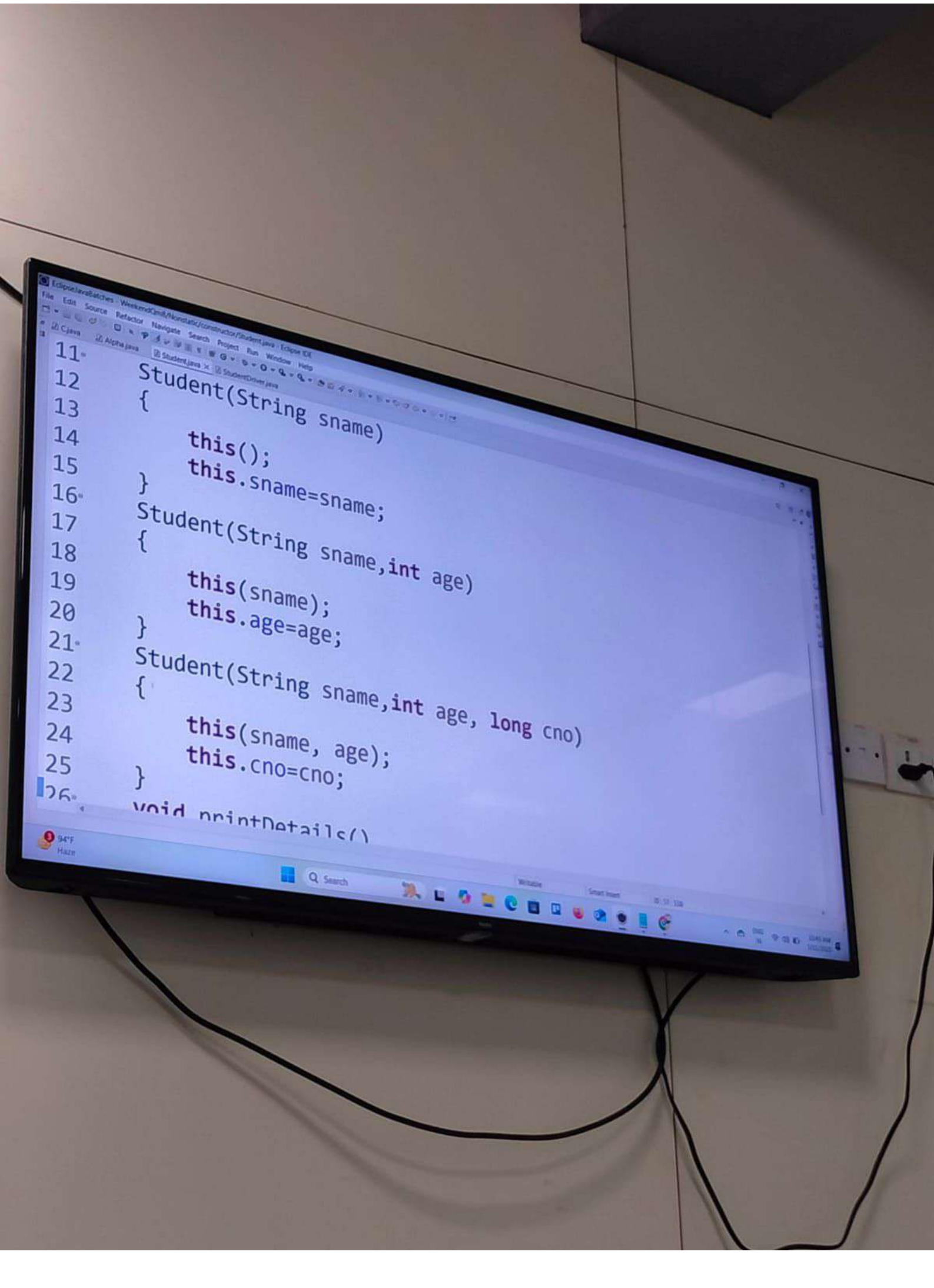
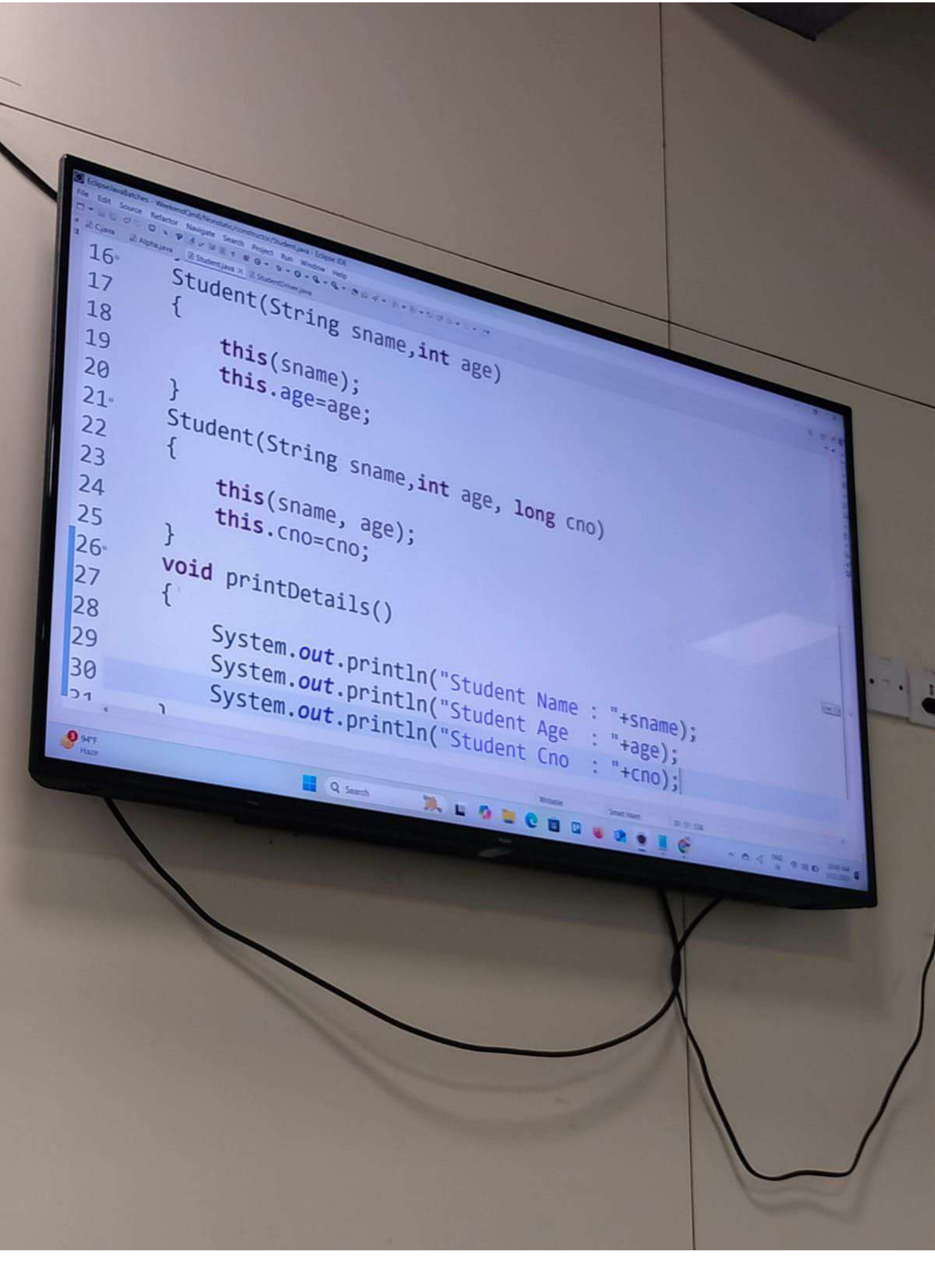
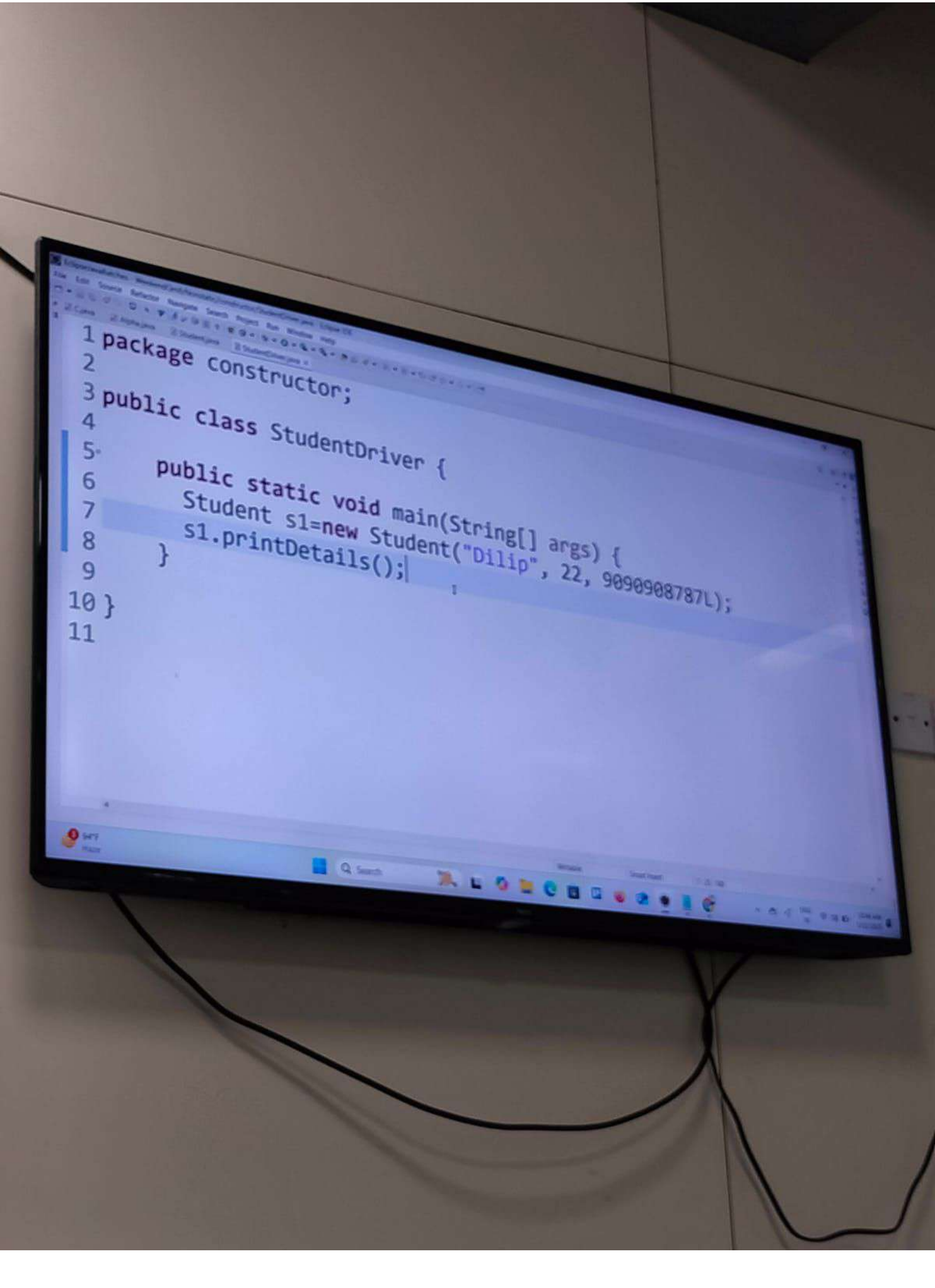

class Alpha {
Alpha()
{
System.out.println("Empty Args");
}
Alpha(int x)
{
this();
System.out.println("int x = "+x);
}
Alpha(int x,int y)
{
this(x);
System.out.println("int y = "+y);
}

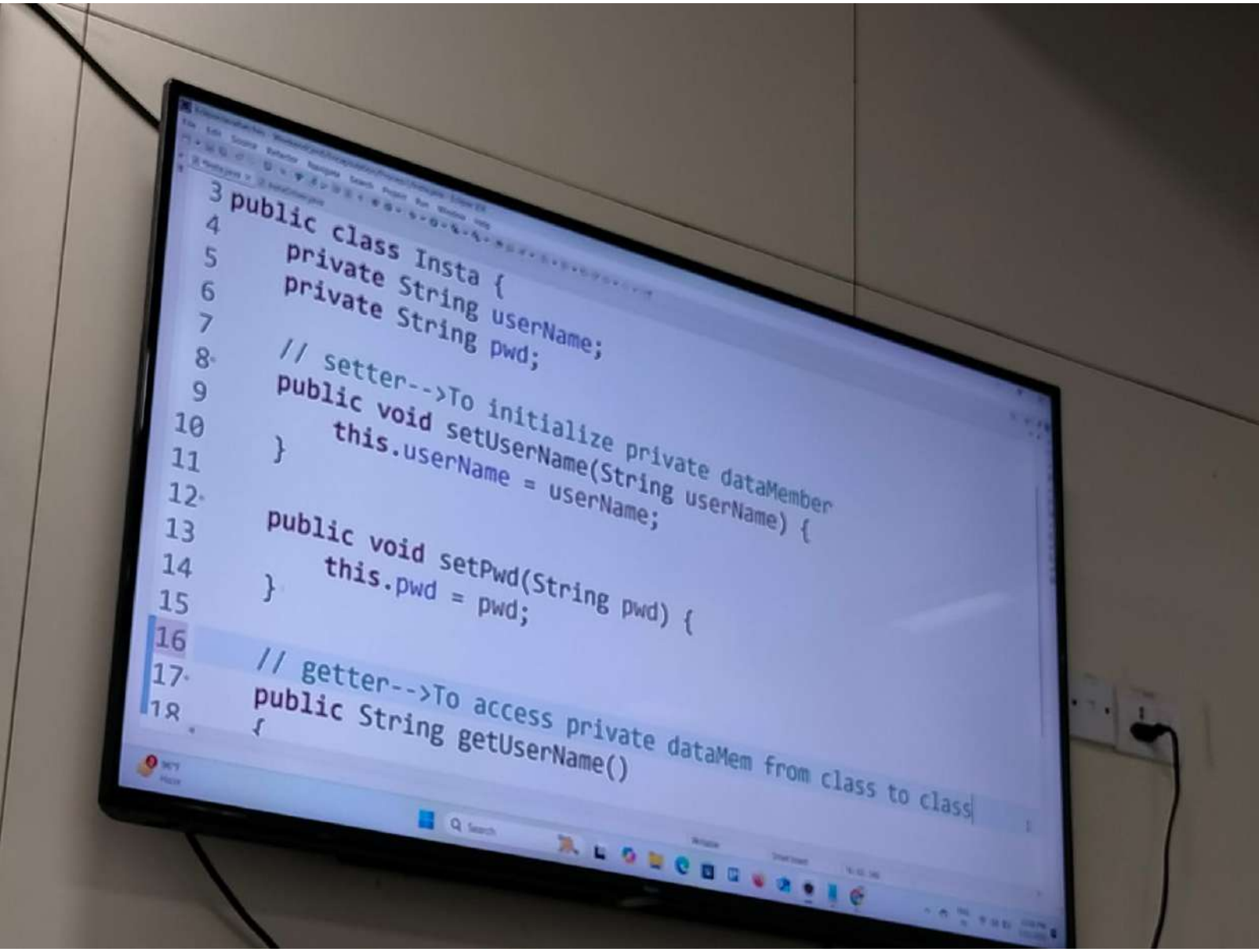

}
Alpha(int x)
{
this();
System.out.println("int x = "+x);
}
Alpha(int x,int y)
{
this(x);
System.out.println("int y = "+y);
}
public static void main(String[] args) {
Alpha a1=new Alpha(10,20);
}
}

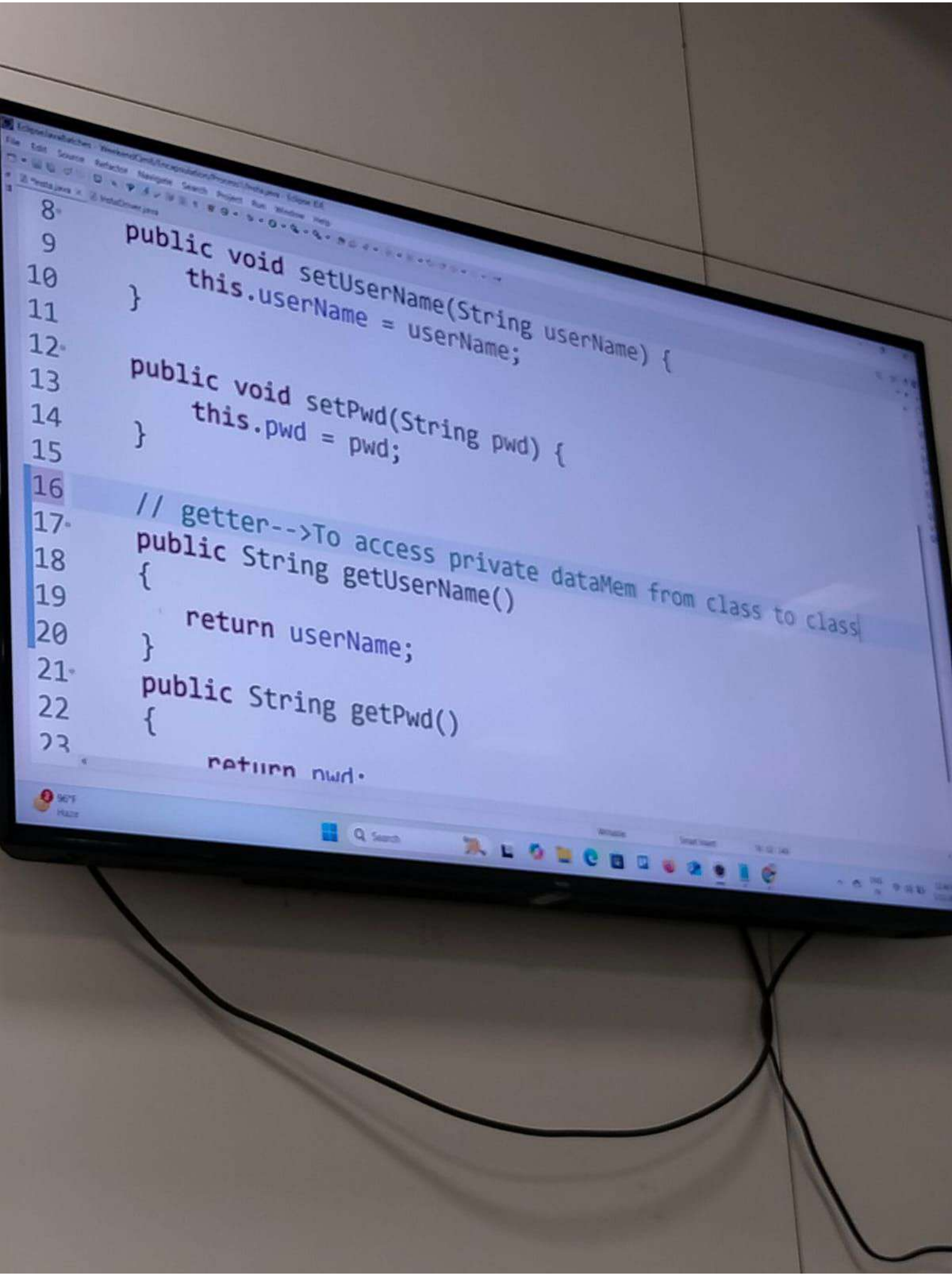

1 package constructor;
2
3 public class Student {
4 String sname;
5 int age;
6 long cno;
7
8 Student() {
9 System.out.println("Student Details");
10 }
11 Student(String sname)
12 {
13 this();
14 this.sname=sname;
15 }
16 Student(String sname,int age)

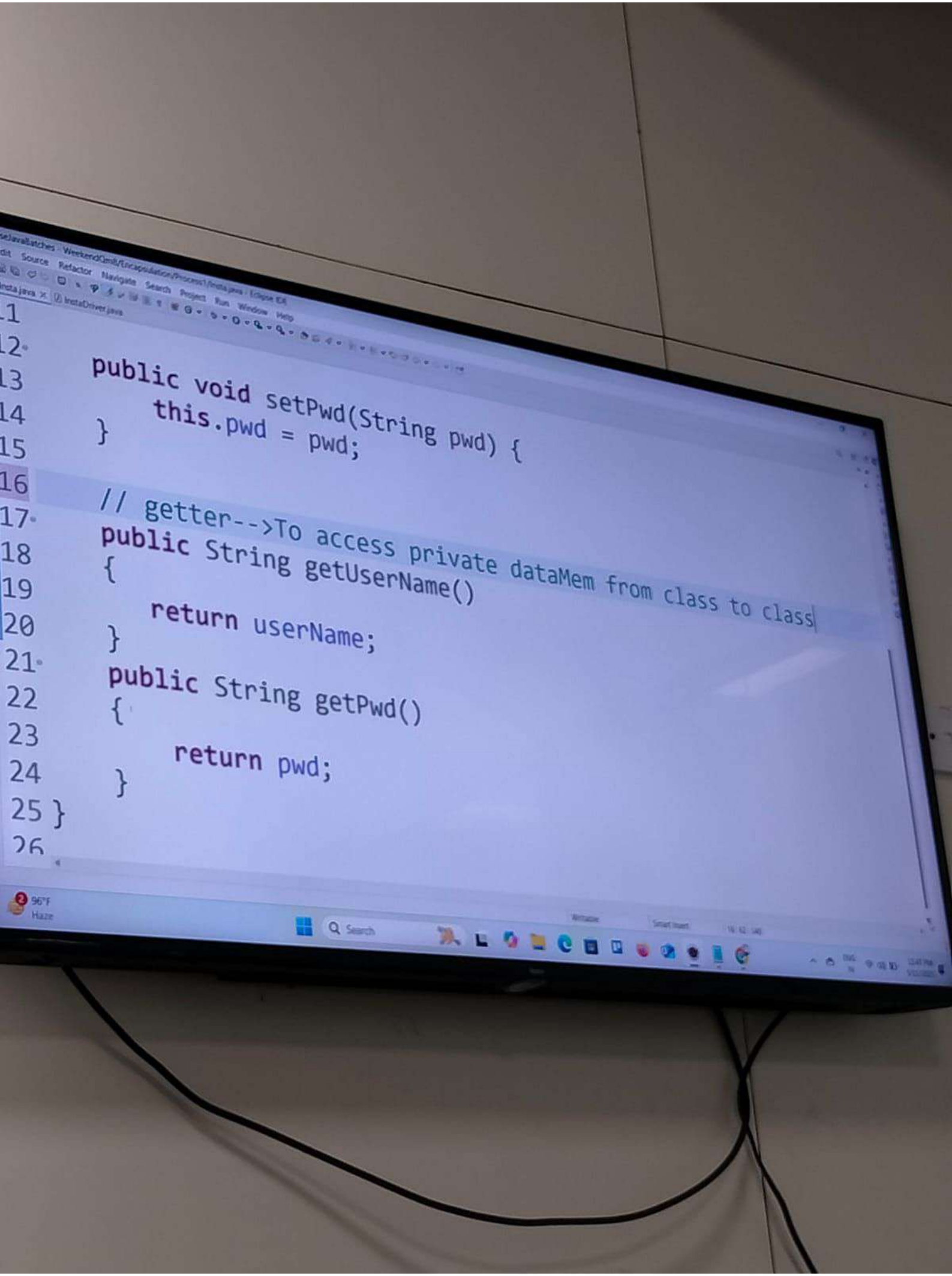

Student(String sname)
{
this();
this.sname=sname;
}
Student(String sname,int age)
{
this(sname);
this.age=age;
}
Student(String sname,int age, long cno)
{
this(sname, age);
this.cno=cno;
}

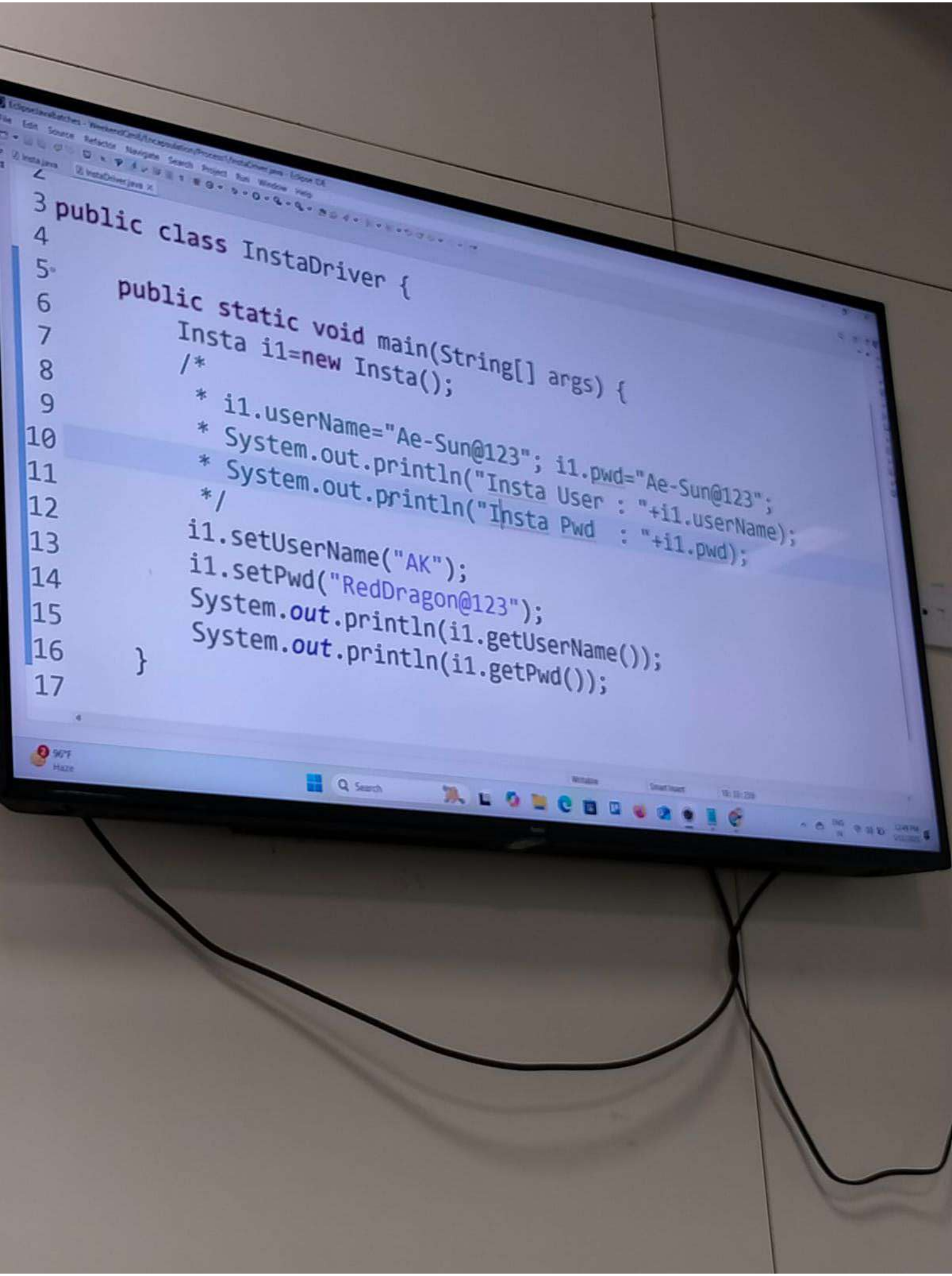

Student(String sname,int age)
{
this(sname);
this.age=age;
}
Student(String sname,int age, long cno)
{
this(sname, age);
this.cno=cno;
}
void printDetails()
{
System.out.println("Student Name : "+sname);
System.out.println("Student Age : "+age);
System.out.println("Student Cno : "+cno);

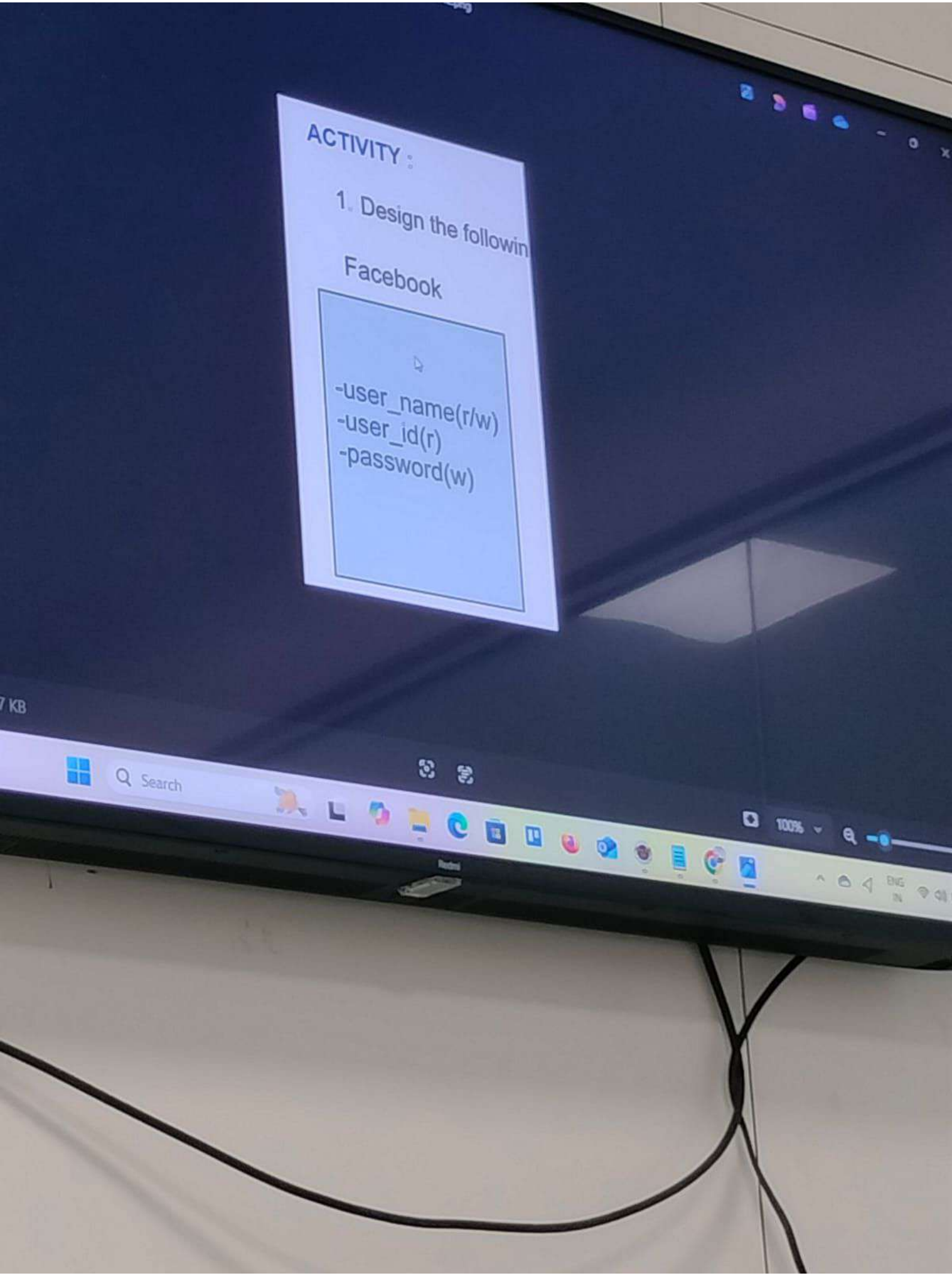

1 package constructor;
2
3 public class StudentDriver {
4
5
6 public static void main(String[] args) {
7 Student s1=new Student("Dilip", 22, 9090908787L);
8 s1.printDetails();
9 }
10 }
11

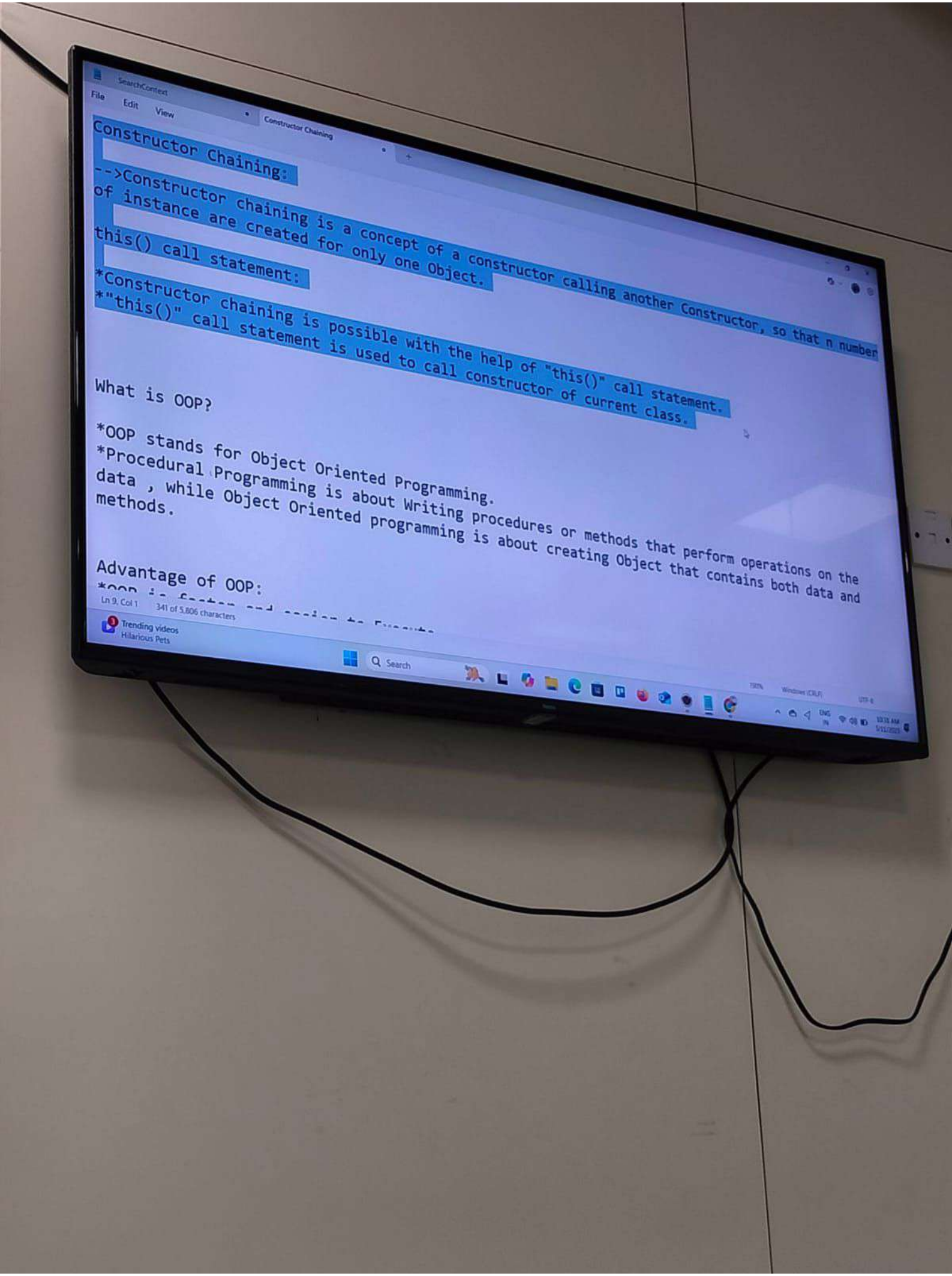

3 public class Insta {
4
5 private String userName;
6 private String pwd;
7
8 // setter-->To initialize private dataMember
9 public void setUserName(String userName) {
10 this.userName = userName;
11 }
12
13 public void setPwd(String pwd) {
14 this.pwd = pwd;
15 }
16
17 // getter-->To access private dataMem from class to class
18 public String getUserName()
{

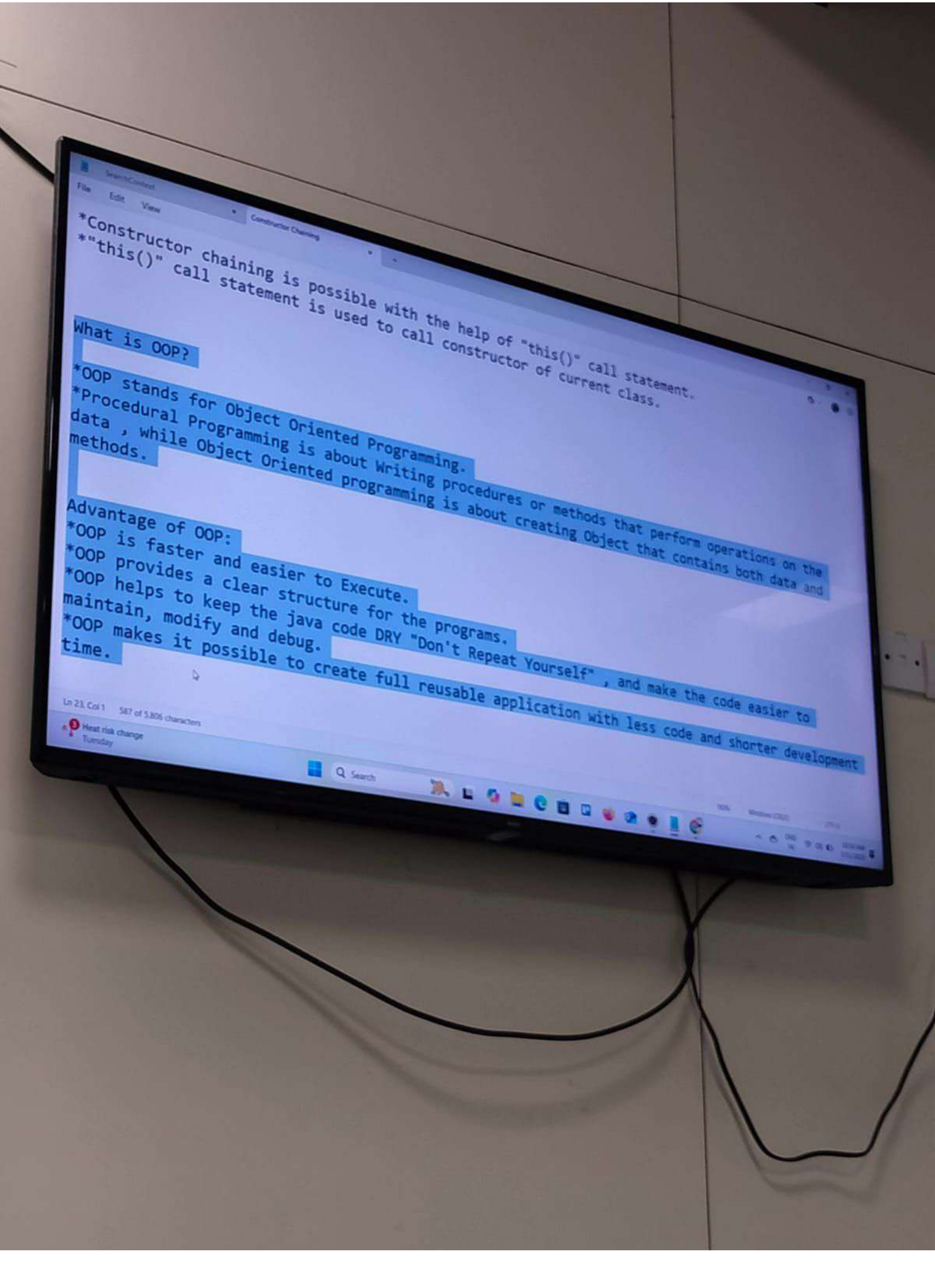

public void setUserName(String userName) {
this.userName = userName;
}
public void setPwd(String pwd) {
this.pwd = pwd;
}
// getter-->To access private dataMem from class to class
public String getUserName()
{
return userName;
}
public String getPwd()
{

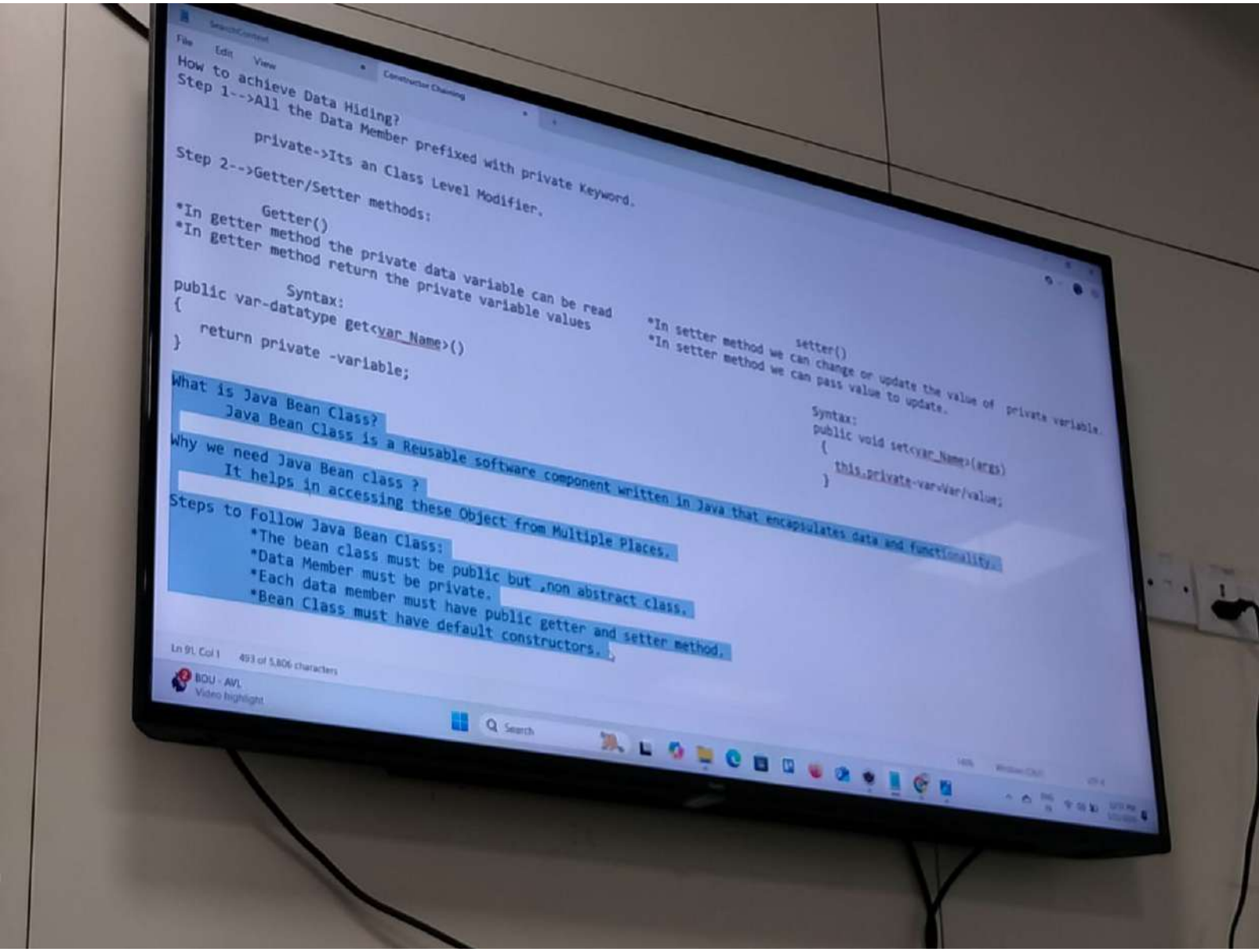

Edit Source Refactor Navigate Search Project Run Window Help
insta.java
InstaDriver.java
public void setPwd(String pwd) {
this.pwd = pwd;
}
// getter-->To access private dataMem from class to class
public String getUserName()
{
return userName;
}
public String getPwd()
{
return pwd;
}
}
96°F
Haze
Search

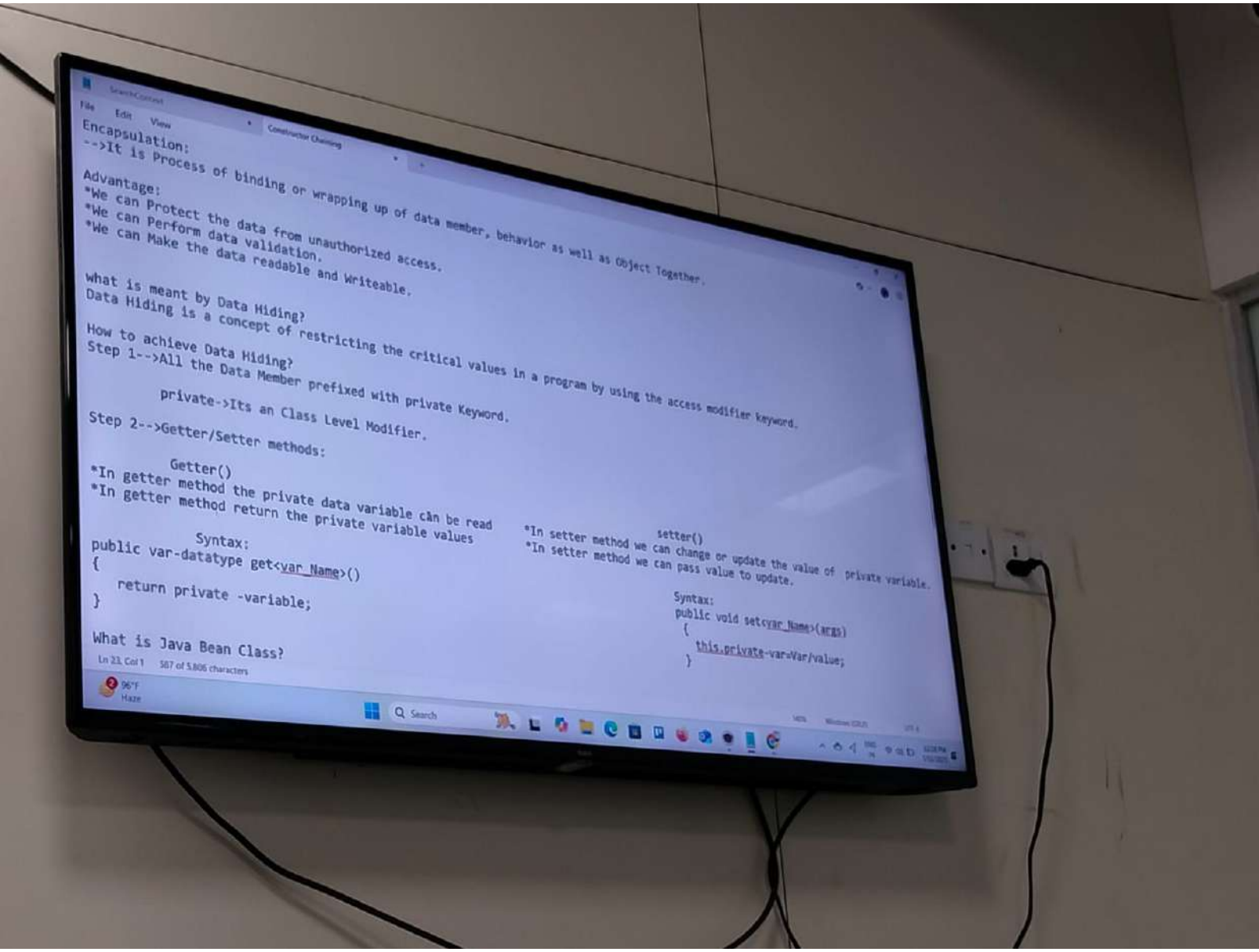

public class InstaDriver {
public static void main(String[] args) {
Insta i1=new Insta();
/*
* i1.userName="Ae-Sun@123"; i1.pwd="Ae-Sun@123";
* System.out.println("Insta User : "+i1.userName);
* System.out.println("Insta Pwd : "+i1.pwd);
*/
i1.setUserName("AK");
i1.setPwd("RedDragon@123");
System.out.println(i1.getUserName());
System.out.println(i1.getPwd());
}

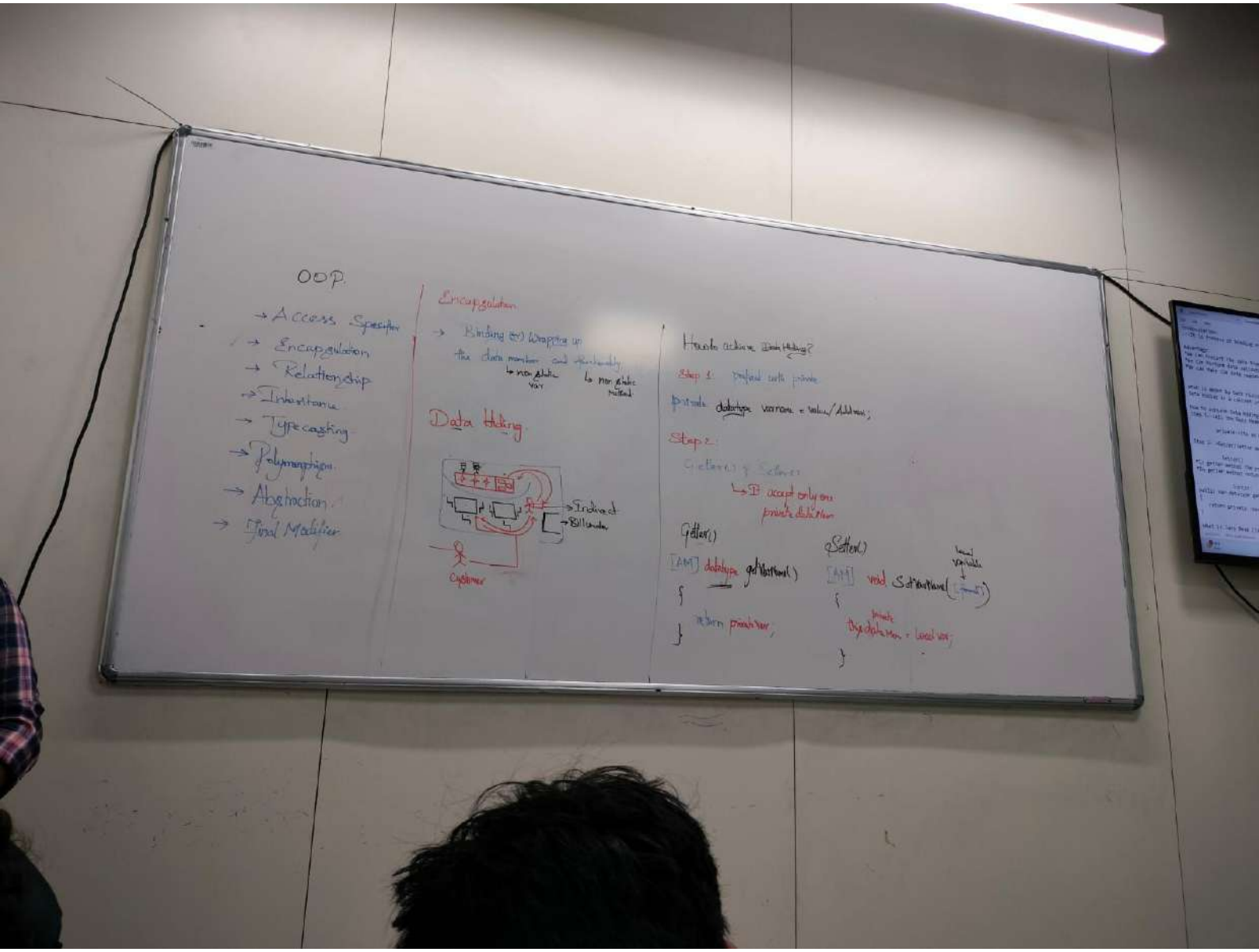

ACTIVITY :
1. Design the followin
Facebook
-user_name(r/w)
-user_id(r)
-password(w)
Search
100%

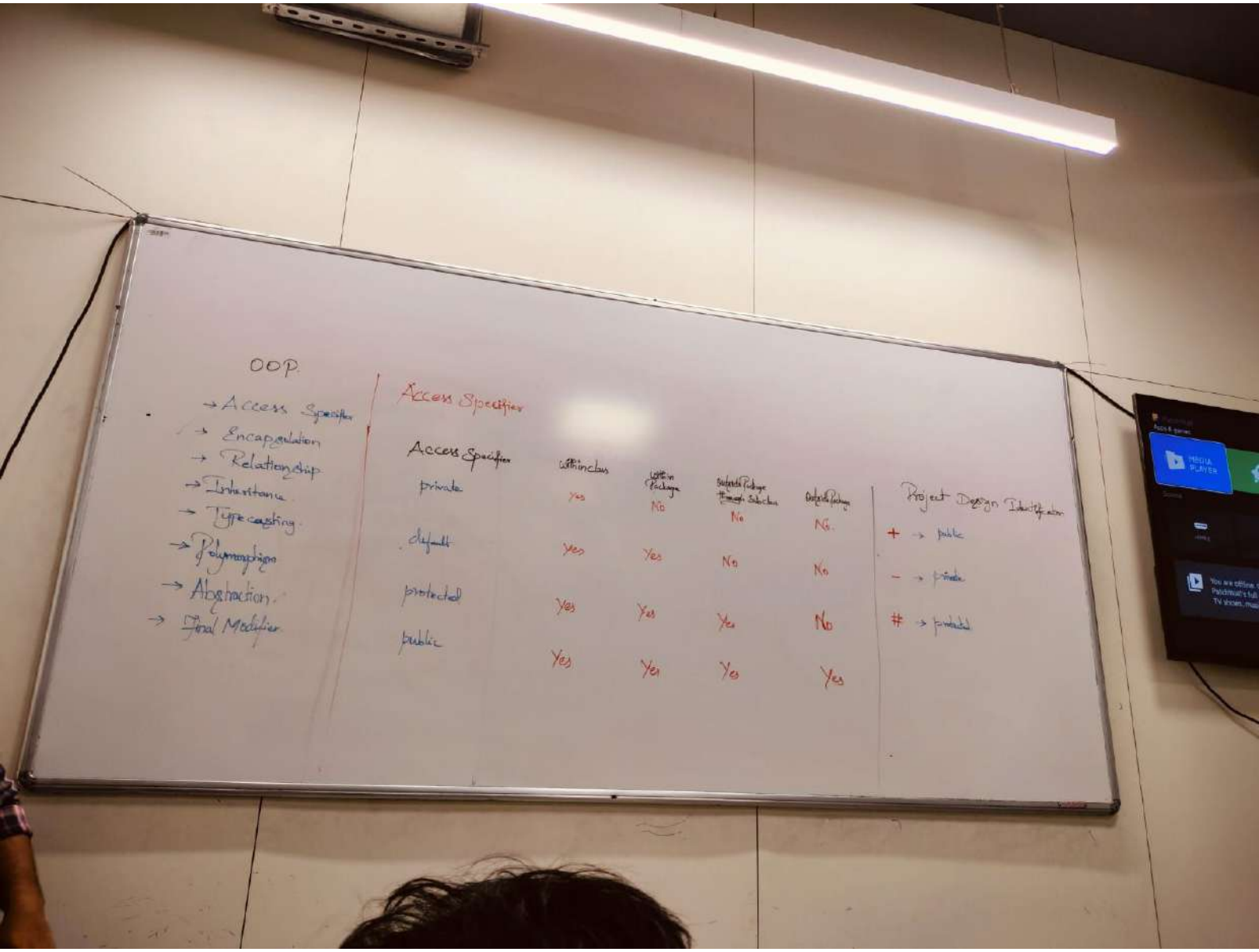

SearchContext
File Edit View
Constructor Chaining
Constructor Chaining:
-->Constructor chaining is a concept of a constructor calling another Constructor, so that n number of instance are created for only one Object.
this() call statement:
*Constructor chaining is possible with the help of "this()" call statement.
*"this()" call statement is used to call constructor of current class.
What is OOP?
*OOP stands for Object Oriented Programming.
*Procedural Programming is about Writing procedures or methods that perform operations on the data , while Object Oriented programming is about creating Object that contains both data and methods.
Advantage of OOP:
Ln 9, Col 1
341 of 5,806 characters
Trending videos
Hilarious Pets
Search


*Constructor chaining is possible with the help of "this()" call statement.
**"this()" call statement is used to call constructor of current class.
What is OOP?
*OOP stands for Object Oriented Programming.
*Procedural Programming is about Writing procedures or methods that perform operations on the data , while Object Oriented programming is about creating Object that contains both data and methods.
Advantage of OOP:
*OOP is faster and easier to Execute.
*OOP provides a clear structure for the programs.
*OOP helps to keep the java code DRY "Don't Repeat Yourself" , and make the code easier to maintain, modify and debug.
*OOP makes it possible to create full reusable application with less code and shorter development time.


How to achieve Data Hiding?
Step 1-->All the Data Member prefixed with private Keyword.
private->Its an Class Level Modifier.
Step 2-->Getter/Setter methods:
Getter()
*In getter method the private data variable can be read
*In getter method return the private variable values
Syntax:
public var-datatype get<var_Name>()
{
return private -variable;
}
setter()
*In setter method we can change or update the value of private variable.
*In setter method we can pass value to update.
Syntax:
public void set<var_Name>(args)
{
this.private-var=var/value;
}
What is Java Bean Class?
Java Bean Class is a Reusable software component written in Java that encapsulates data and functionality.
Why we need Java Bean class ?
It helps in accessing these Object from Multiple Places.
Steps to Follow Java Bean Class:
*The bean class must be public but ,non abstract class.
*Data Member must be private.
*Each data member must have public getter and setter method.
*Bean Class must have default constructors.

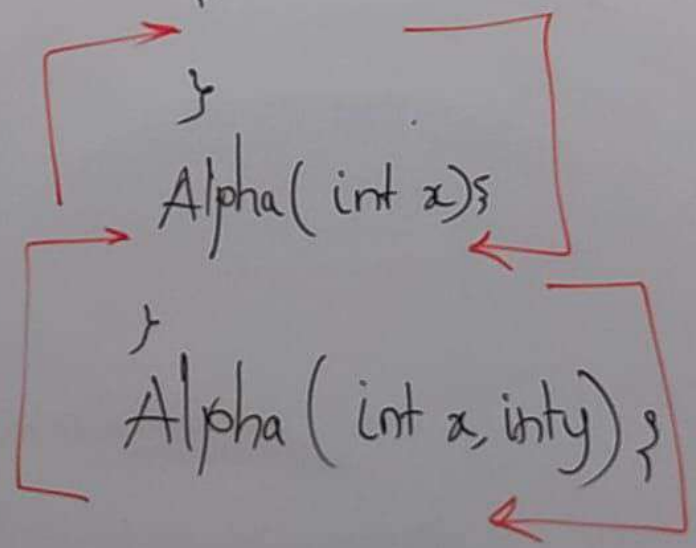

Encapsulation:
-->It is Process of binding or wrapping up of data member, behavior as well as Object Together.
Advantage:
*We can Protect the data from unauthorized access.
*We can Perform data validation.
*We can Make the data readable and Writeable.
what is meant by Data Hiding?
Data Hiding is a concept of restricting the critical values in a program by using the access modifier keyword.
How to achieve Data Hiding?
Step 1-->All the Data Member prefixed with private Keyword.
private->Its an Class Level Modifier.
Step 2-->Getter/Setter methods:
Getter()
*In getter method the private data variable can be read
*In getter method return the private variable values
Syntax:
public var-datatype get<var_Name>()
{
return private -variable;
}
setter()
*In setter method we can change or update the value of private variable.
*In setter method we can pass value to update.
Syntax:
public void set<var_Name>(ARGS)
{
this.private-var=Var/value;
}
What is Java Bean Class?


OOP
→ Access Specifier
→ Encapsulation
→ Relationship
→ Inheritance
→ Typecasting
→ Polymorphism
→ Abstraction
→ Final Modifier
Encapsulation
→ Binding (or) Wrapping up the data member and functionality
to non-static var
to non-static method
Data Hiding
Indirect
Billmachine
Customer
How to achieve Data Hiding?
Step 1: Default scope private
private datatype varname = value/defaultvalue;
Step 2:
Getters & Setters
↳ It accept only one private data item
Getter()
[AM] datatype getVariable()
{
return privatevar;
}
Setter()
local variable
[AM] void setVariable(datatype)
{
private datatype = localvar;
}


OOP
→ Access Specifier
→ Encapsulation
→ Relationship
→ Inheritance
→ Type casting
→ Polymorphism
→ Abstraction
→ Final Modifier
Access Specifier
Access Specifier
within class
Yes
private
default
protected
public
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
No
No
No
No
No
No
Project Design Identification
+ → public
- → private
→ protected

Constructor chaining

```
class Alpha
{
    Alpha()
    {
    }
    Alpha(int x);
    {
    }
    Alpha(int x, int y);{
    }
}
```



this() call statement