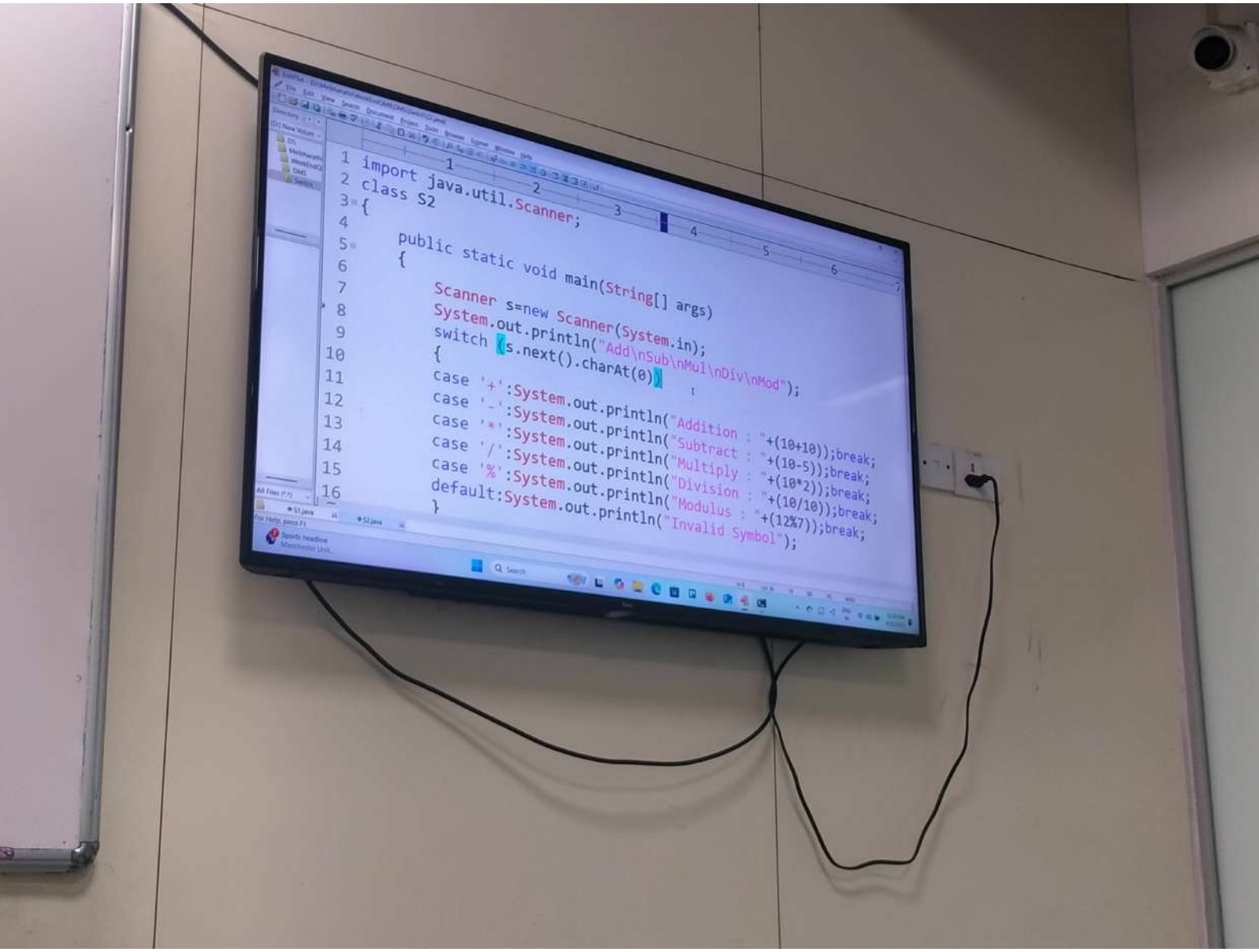
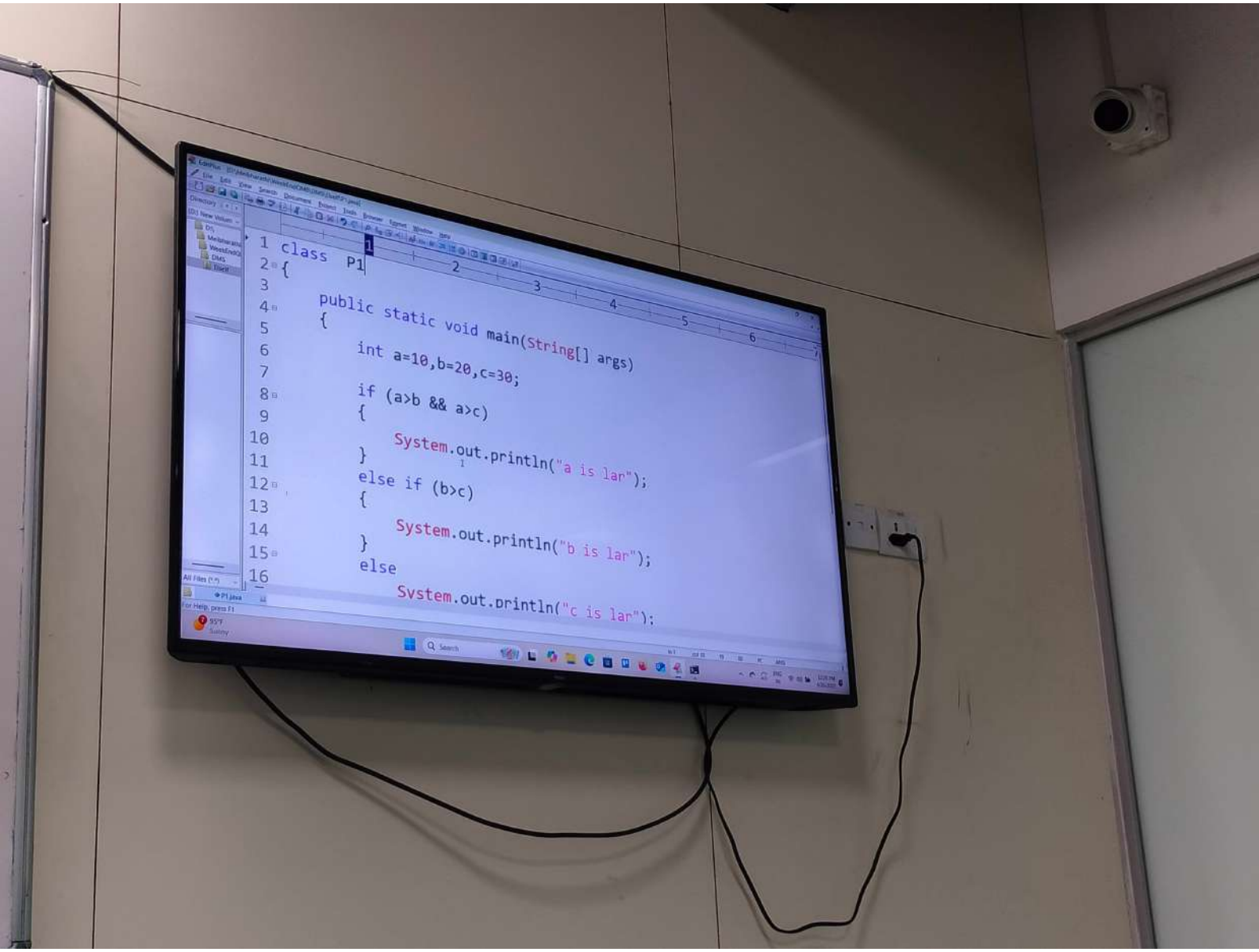
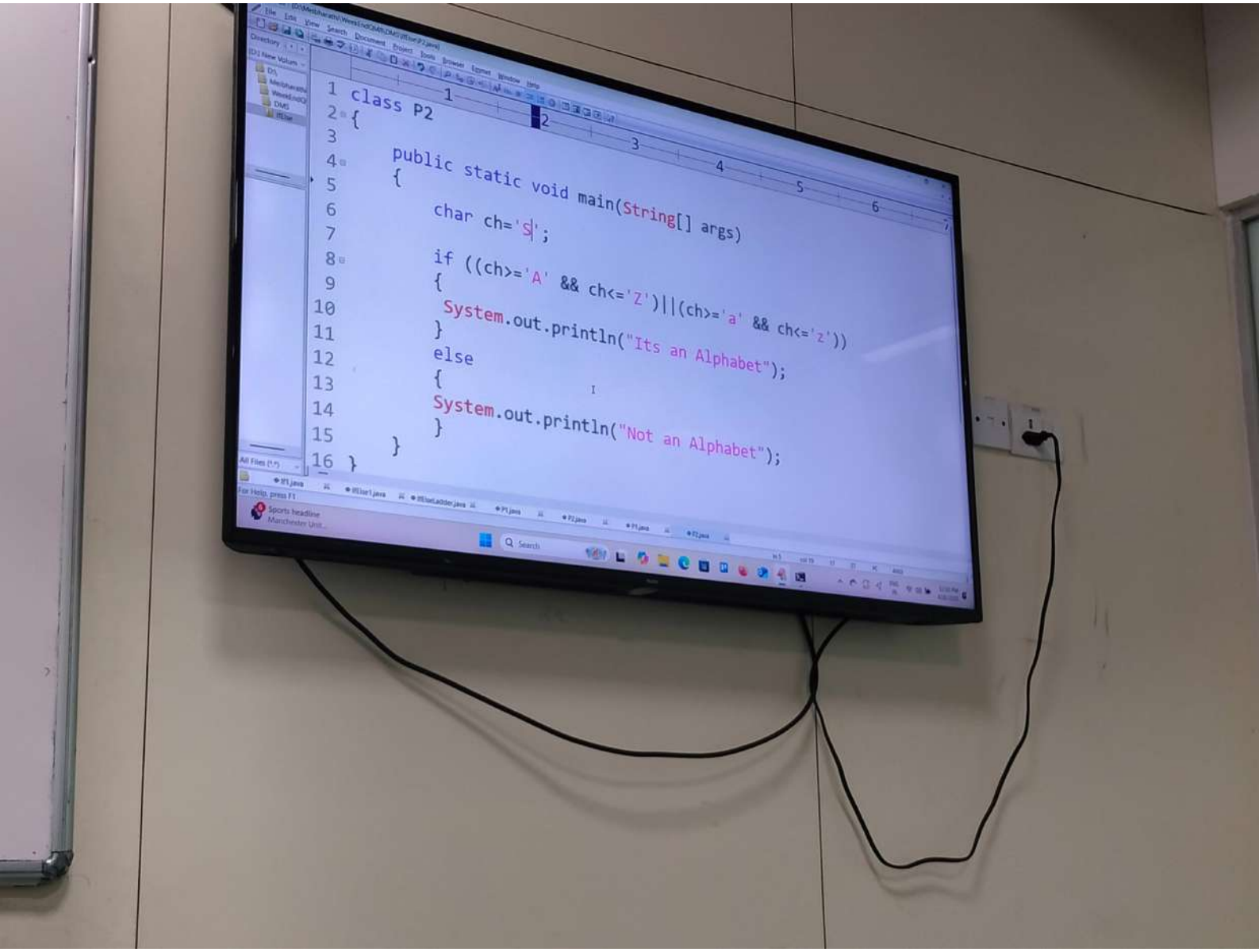
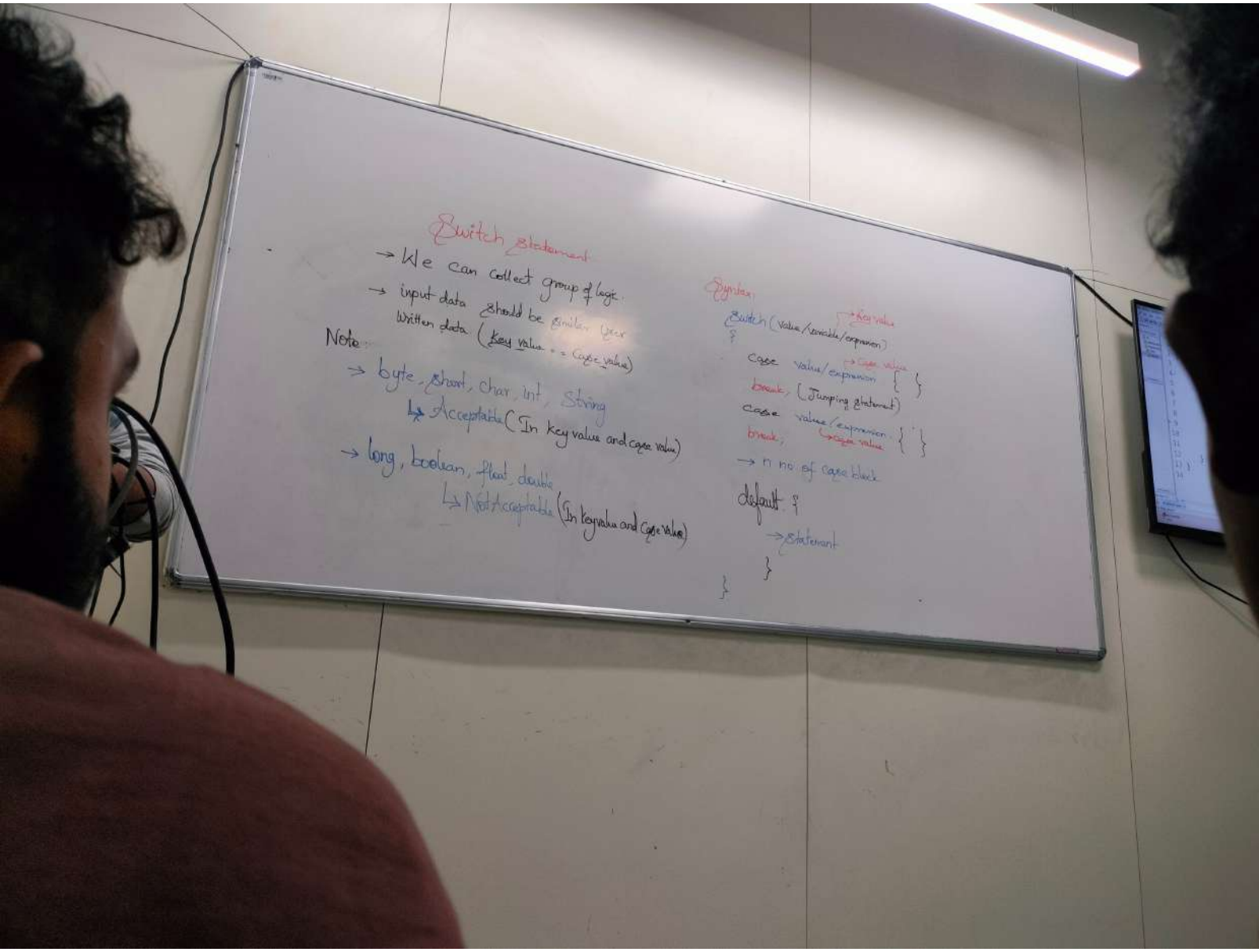
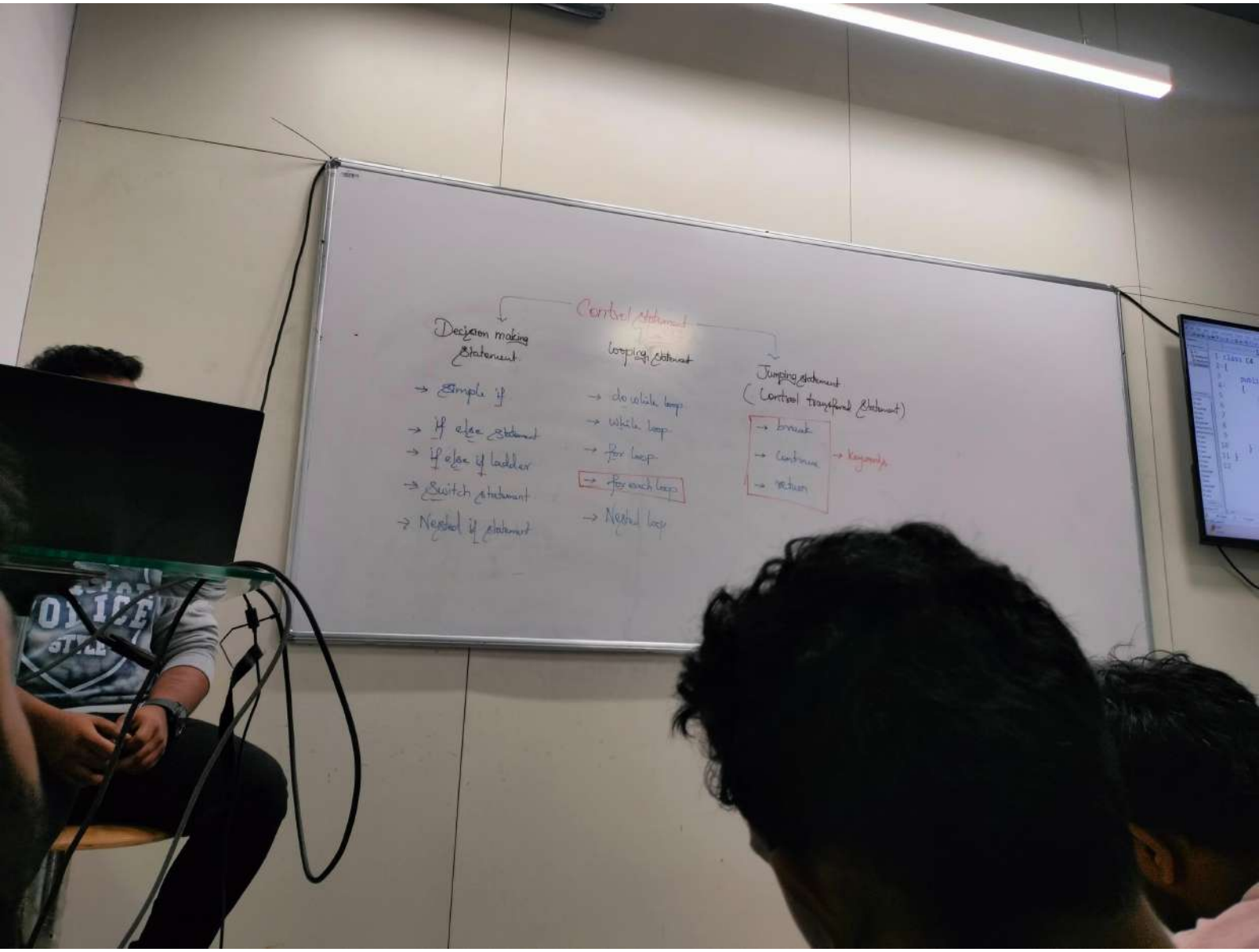
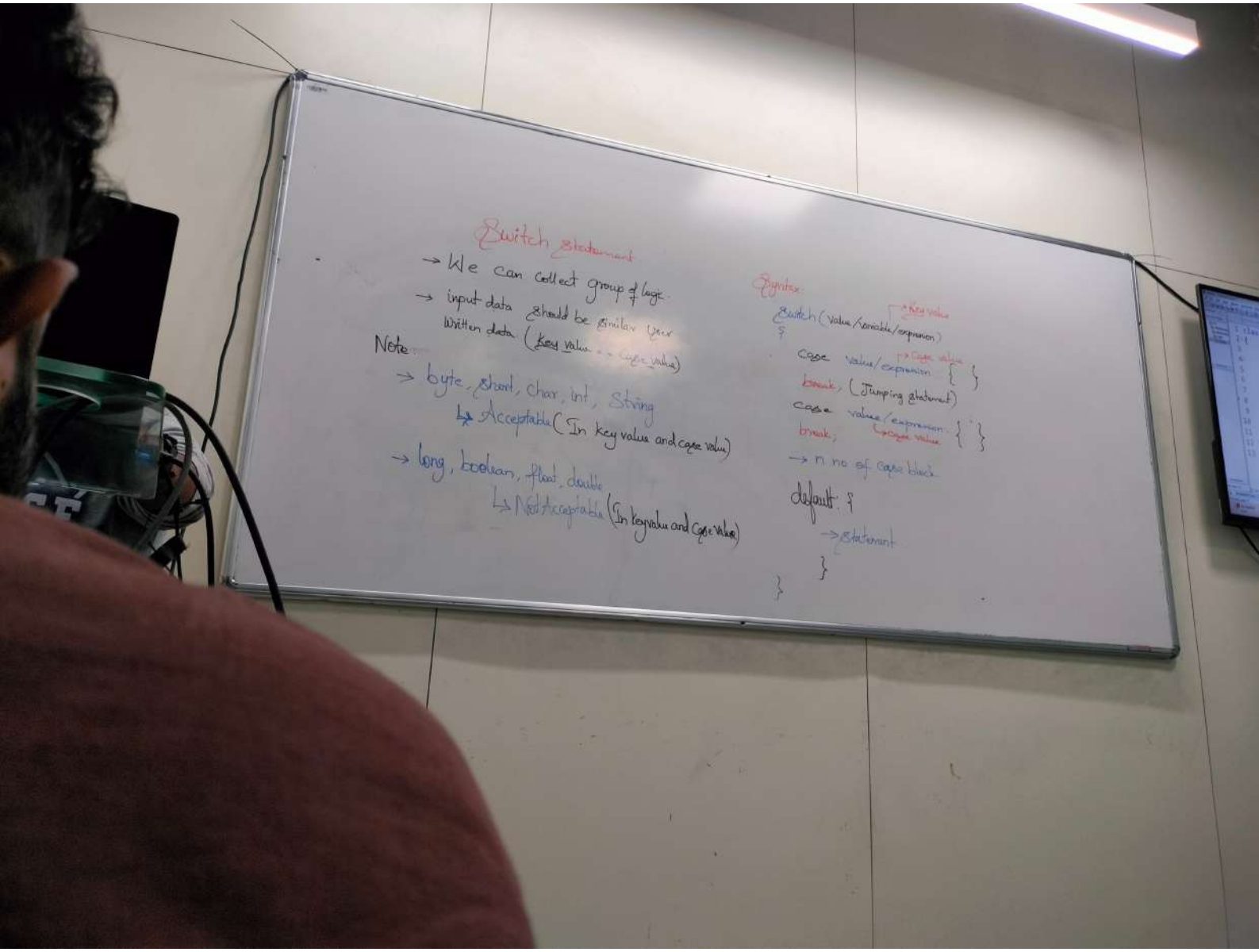

import java.util.Scanner;
class S2
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.println("Add\nSub\nMul\nDiv\nMod");
switch (s.next().charAt(0))
{
case '+':System.out.println("Addition : "+(10+10));break;
case '-':System.out.println("Subtract : "+(10-5));break;
case '*':System.out.println("Multiply : "+(10*2));break;
case '/':System.out.println("Division : "+(10/10));break;
case '%':System.out.println("Modulus : "+(12%7));break;
default:System.out.println("Invalid Symbol");
}


class P1
{
public static void main(String[] args)
{
int a=10,b=20,c=30;
if (a>b && a>c)
{
System.out.println("a is lar");
}
else if (b>c)
{
System.out.println("b is lar");
}
else
System.out.println("c is lar");


class P2
{
public static void main(String[] args)
{
char ch='S';
if ((ch>='A' && ch<='Z')||(ch>='a' && ch<='z'))
{
System.out.println("Its an Alphabet");
}
else
{
System.out.println("Not an Alphabet");
}
}
}


Switch Statement
→ We can collect group of logic.
→ input data should be similar type
Written data (key value == case value)
Note:
→ byte, short, char, int, String
↳ Acceptable (In key value and case value)
→ long, boolean, float, double
↳ Not Acceptable (In key value and case value)
Syntax:
switch (value/variable/expression)
{
case value/expression :
break; (Jumping statement)
case value/expression :
break;
→ n no. of case block
default :
→ Statement
}
}


Control statement
Decision making statement
→ Simple if
→ If else statement
→ If else if ladder
→ Switch statement
→ Nested if statement
Looping statement
→ do while loop
→ While loop
→ For loop
→ for each loop
→ Nested loop
Jumping statement
(Control transfer statement)
→ break
→ continue
→ return
→ keywords


Switch statement
→ We can collect group of logic
→ input data should be similar year
written data (key value = = case value)
Note:
→ byte, short, char, int, String
↳ Acceptable (In key value and case value)
→ long, boolean, float, double
↳ Not Acceptable (In key value and case value)
Syntax:
switch (value/variable/expression)
{
case value/expression { }
break; (Jumping statement)
case value/expression { }
break;
→ n no of case block
default: {
→ statement
}
}