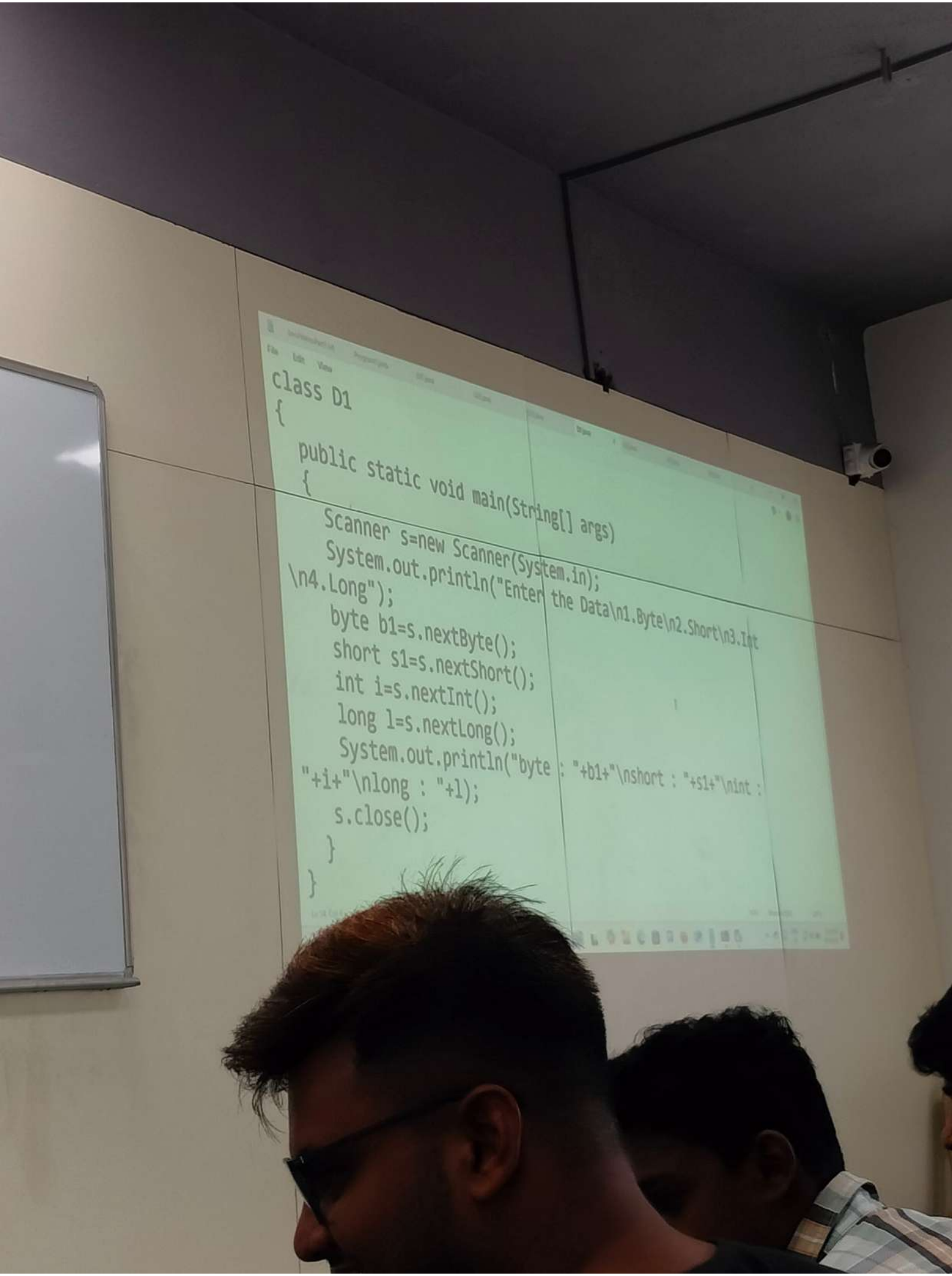
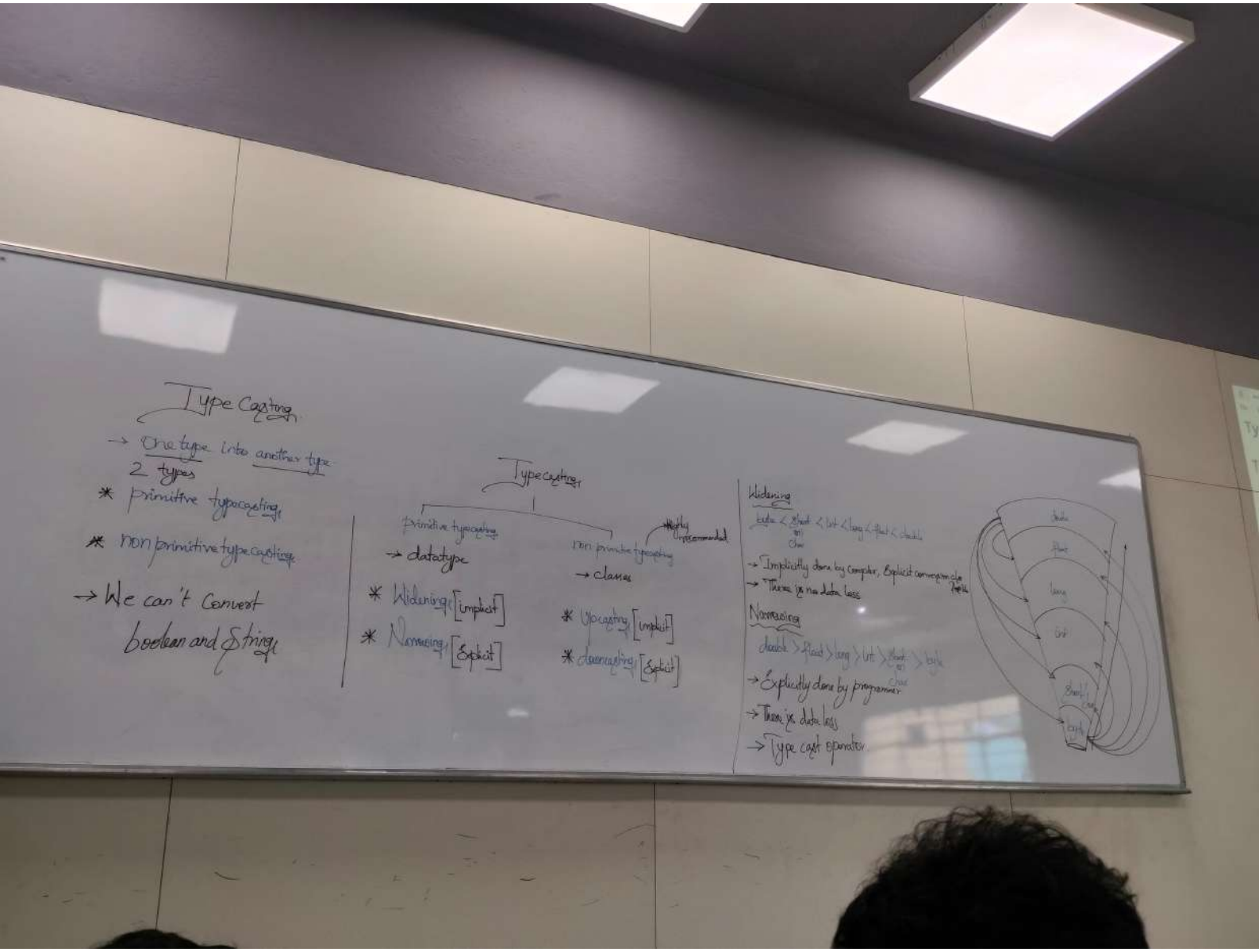
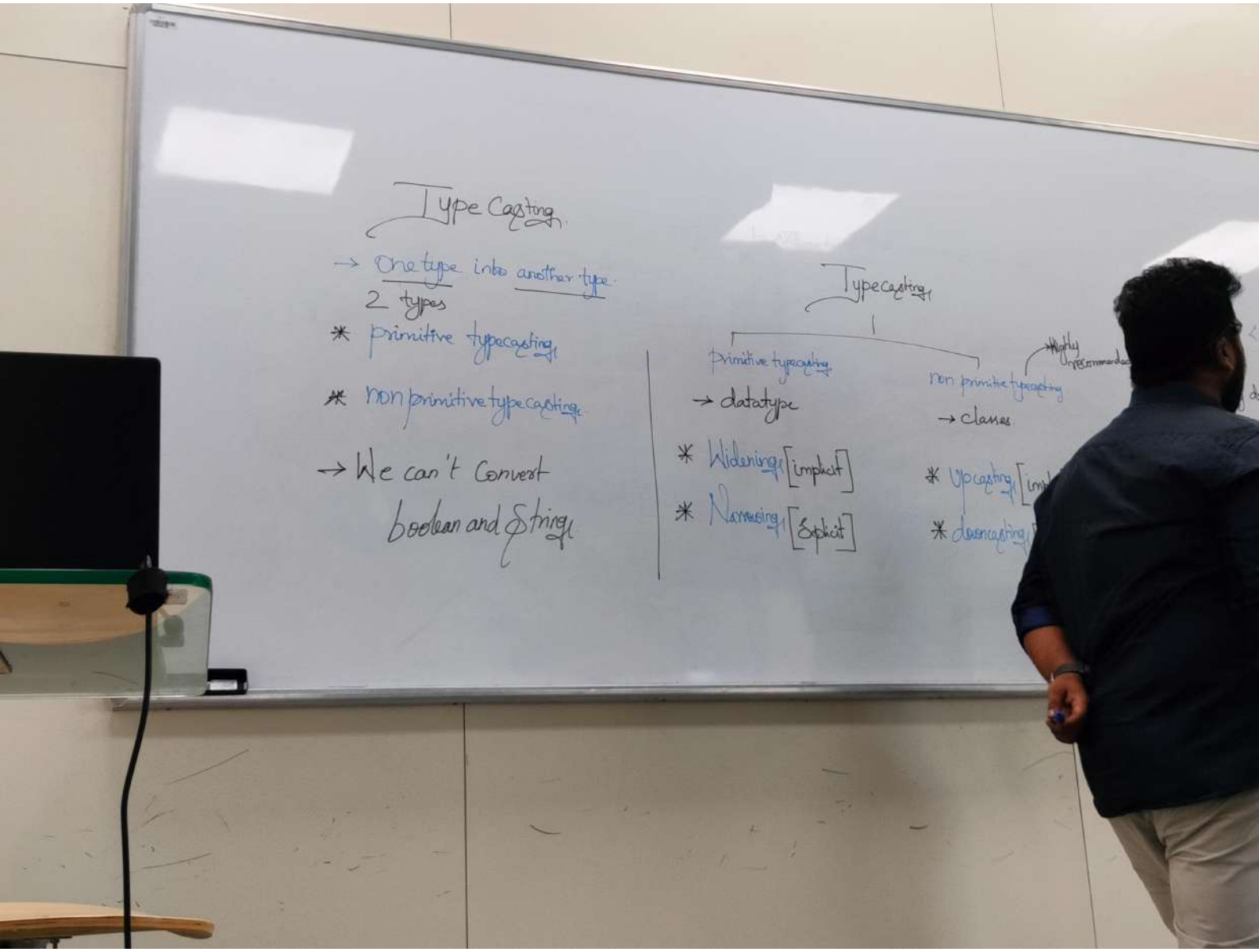
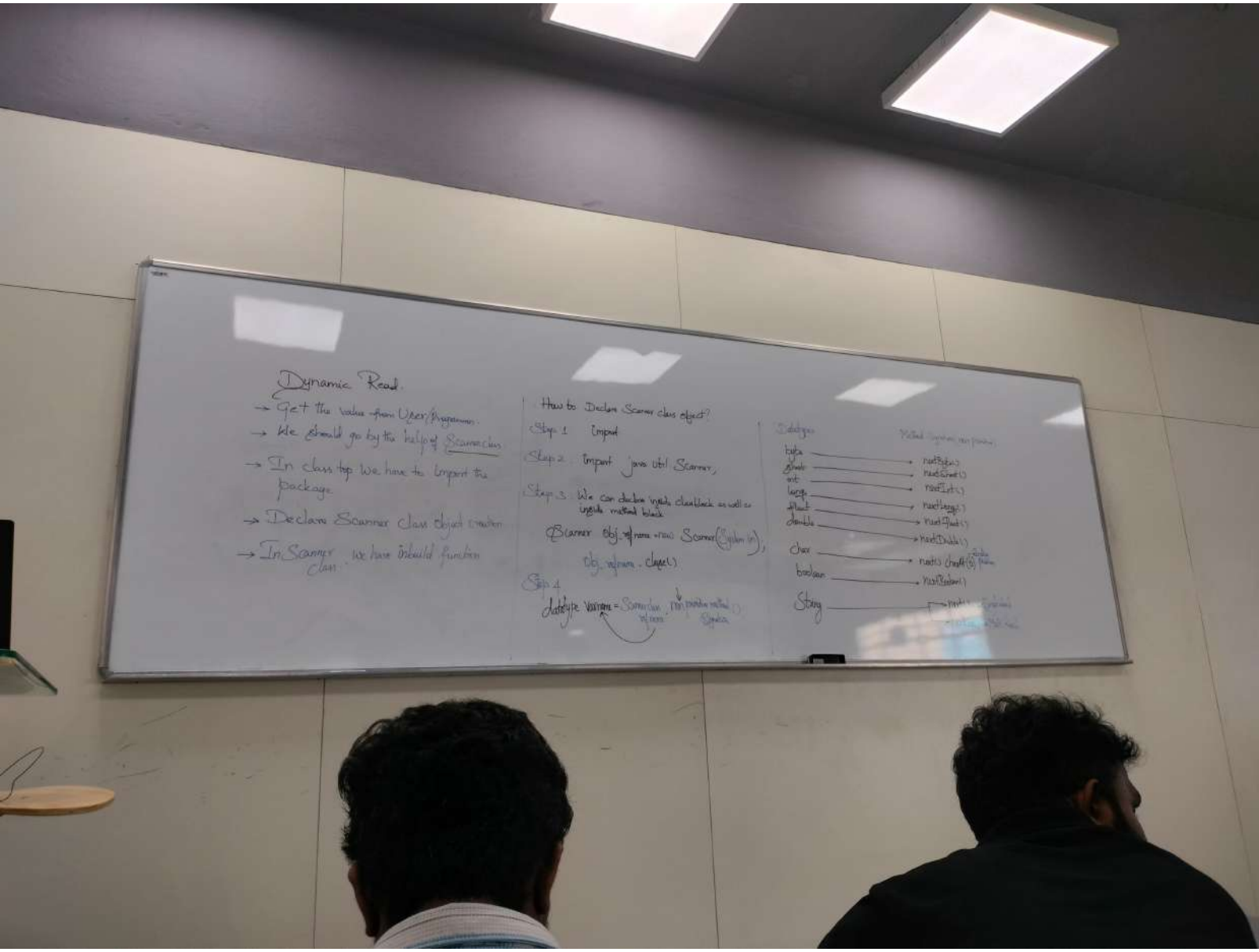

class D1
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.println("Enter the Data\n1.Byte\n2.Short\n3.Int
\n4.Long");
byte b1=s.nextByte();
short s1=s.nextShort();
int i=s.nextInt();
long l=s.nextLong();
System.out.println("byte : "+b1+"\nshort : "+s1+"\nint :
"+i+"\nlong : "+l);
s.close();
}
}


Type Casting
→ One type into another type
2 types
* primitive typecasting
** non primitive typecasting
→ We can't Convert boolean and String
Typecasting
Primitive typecasting
→ datatype
* Widening [implicit]
* Narrowing [Explicit]
Non primitive typecasting
Highly recommended
→ class
* Upcasting [implicit]
* downcasting [Explicit]
Widening
byte < short < int < long < float < double
char
→ Implicitly done by Compiler, Explicit conversion also
→ There is no data loss
Narrowing
double > float > long > int > short > byte
char
→ Explicitly done by programmer
→ There is data loss
→ Type cast operator
double
float
long
int
short
byte
char


Type Casting
→ One type into another type.
2 types
* primitive typecasting
* non primitive typecasting
→ We can't Convert
boolean and String
Typecasting
primitive typecasting
→ datatype
* Widening [implicit]
* Narrowing [Explicit]
non primitive typecasting
→ classes
* upcasting
* downcasting


Dynamic Read.
→ Get the value from User/Programmer.
→ We should go by the help of Scanner class
→ In class top We have to import the package
→ Declare Scanner class object creation
→ In Scanner class We have inbuild function
How to Declare Scanner class object?
Step 1 Import
Step 2 Import java.util.Scanner;
Step 3 We can declare inside class block as well as inside method block
Scanner obj_ref_name = new Scanner(System.in);
obj_ref_name.close();
Step 4
Datatype
byte
short
int
long
float
double
char
boolean
String