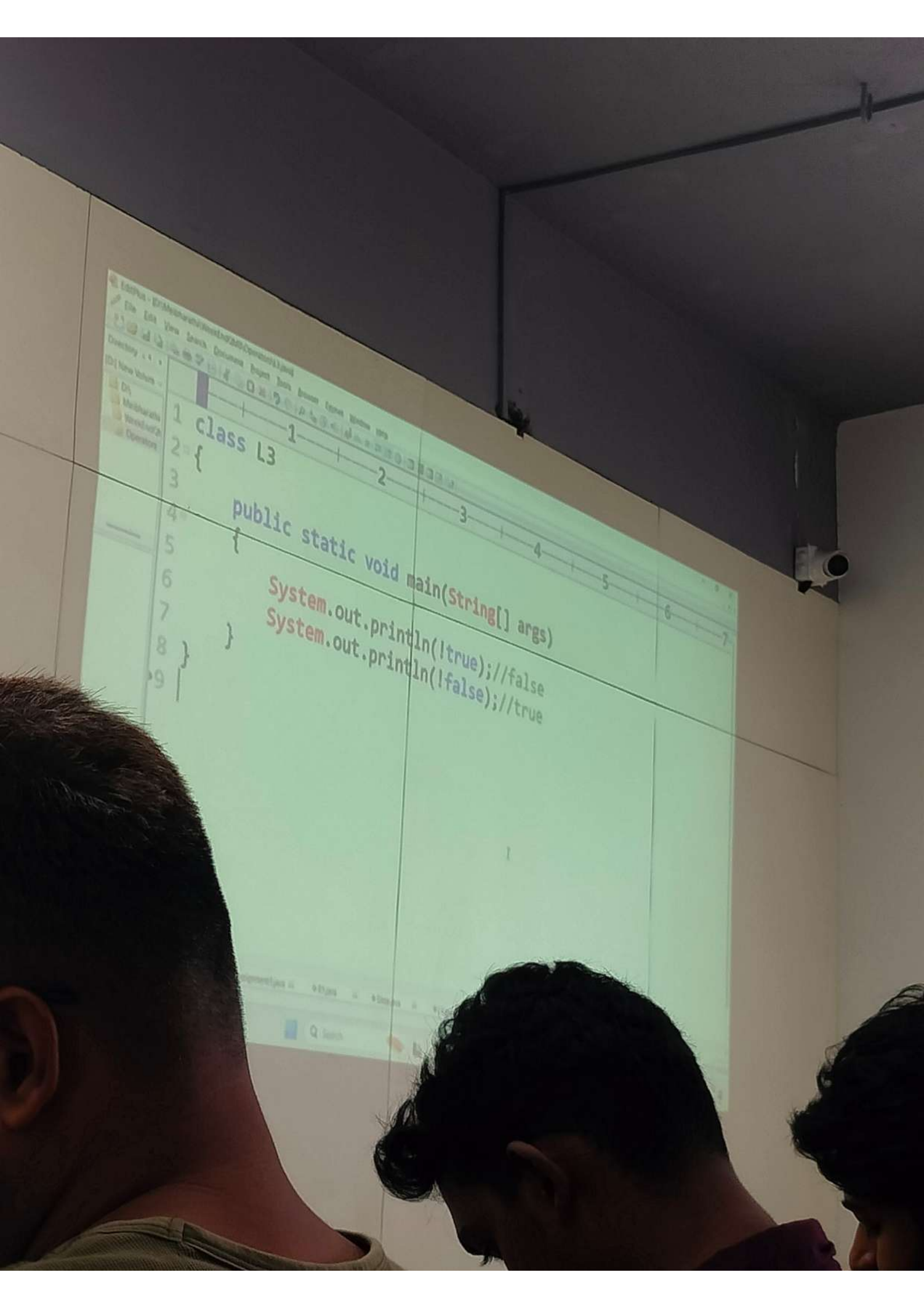


```
1 class L1
2 {
3
4     public static void main(String[] args)
5     {
6         String shoe="White Sneaker";
7         double price=1000;
8
9         boolean afford=(shoe=="White Sneaker") && (price<=1000);
10
11         System.out.println("Is it possible to Buy : "+afford);
12     }
13 }
```

```
1 class L2
2 {
3
4     public static void main(String[] args)
5     {
6         String food="GobiManchurian";
7         double price=290;
8
9         boolean afford=(food=="GobiManchurian") && (price<=190);
10
11         System.out.println("Is it possible to Buy : "+afford);
12     }
13 }
```

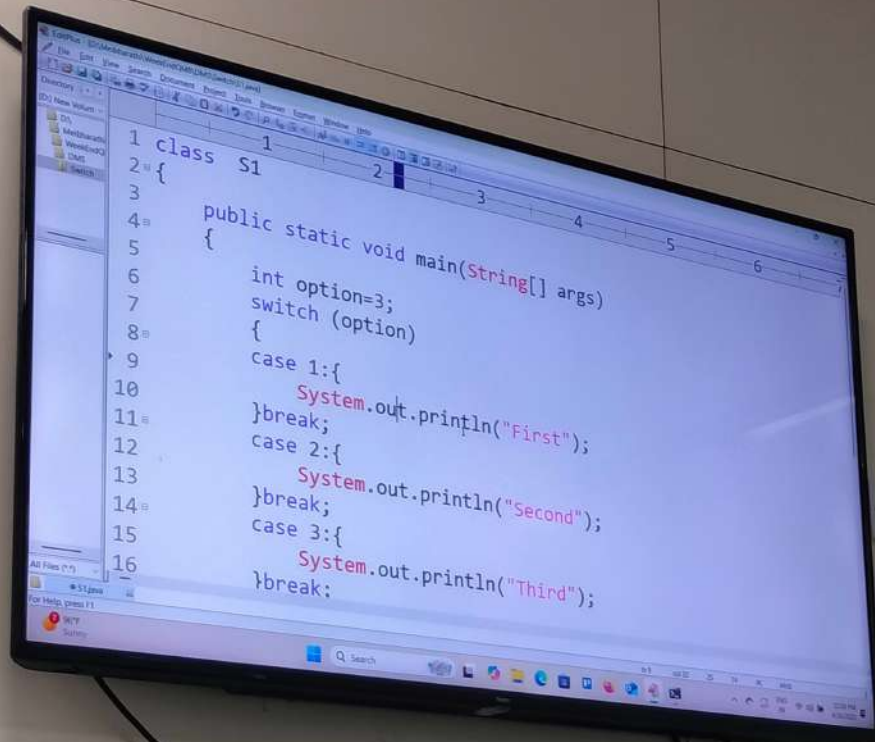


```
1 class L3
2 {
3
4     public static void main(String[] args)
5     {
6         System.out.println(!true); //false
7         System.out.println(!false); //true
8     }
9 }
```

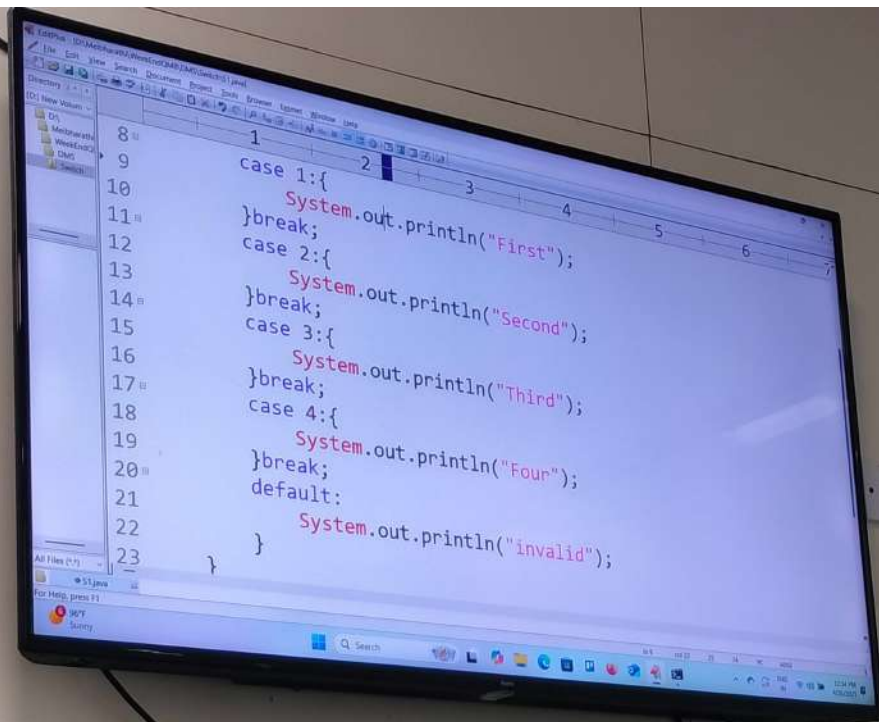
```
1 class Shop
2 {
3
4     public static void main(String[] args)
5     {
6         double shoe1=3250.50,shoe2=1890.0;
7         double wallet1=3100.00,wallet2=2000;
8         //check the Condition
9         boolean afford1=(shoe1 <= wallet1);//false
10        boolean afford2=(shoe2 <= wallet2);//true
11
12        System.out.println("Affordable price for shoe1 "+afford1);
13        System.out.println("Affordable price for shoe2 "+afford2);
14    }
15 }
16
```



```
1 class IfElse1
2 {
3     public static void main(String[] args)
4     {
5         //Syntax:
6         if (/*Condition*/)
7         //Either true or false
8         {
9             //true : block has to Execute
10        }
11        //false :if block has to Skip
12        else
13        {
14            //else block has to Execute
15        }
16    }
}
```



```
1 class S1
2 {
3     public static void main(String[] args)
4     {
5         int option=3;
6         switch (option)
7         {
8             case 1:{
9                 System.out.println("First");
10            }break;
11            case 2:{
12                System.out.println("Second");
13            }break;
14            case 3:{
15                System.out.println("Third");
16            }break;
```

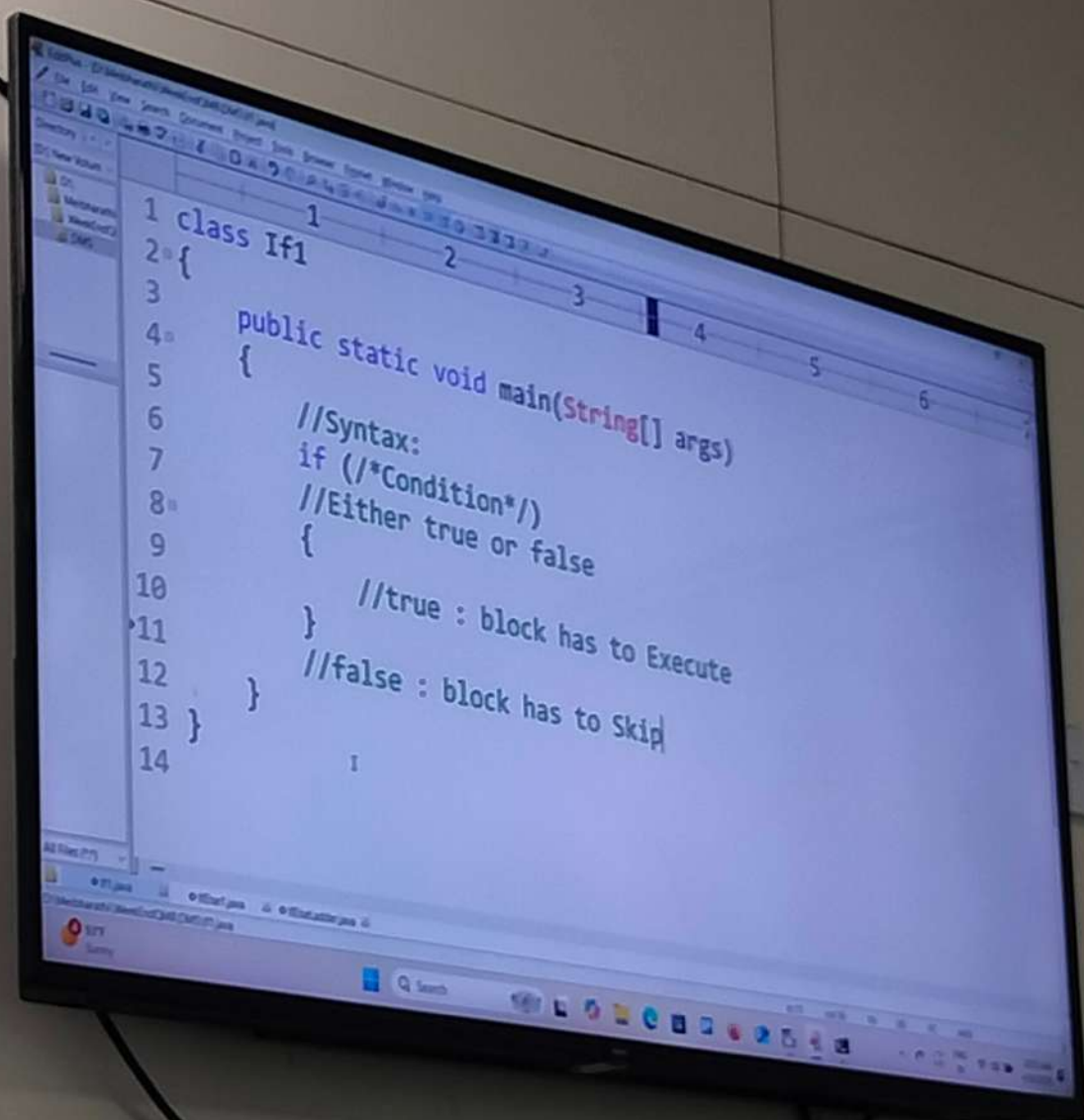


```
8 | 1
9 | case 1:{
10 |     System.out.println("First");
11 | }break;
12 | case 2:{
13 |     System.out.println("Second");
14 | }break;
15 | case 3:{
16 |     System.out.println("Third");
17 | }break;
18 | case 4:{
19 |     System.out.println("Four");
20 | }break;
21 | default:
22 |     System.out.println("invalid");
23 | }
```

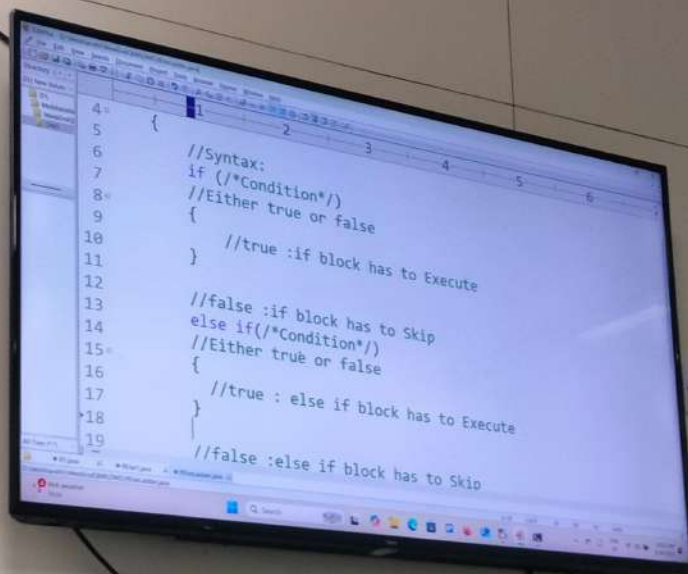
```
EdPlus - D:\Maharathi\Weekend\Assignment1.java
File Edit View Search Document Project Tools Browser Format Window Help
Directory: D:\New Volume
En Maharathi Weekend Operators
1
2 {
3
4 public static void main(String[] args)
5 {
6     int a=10;
7
8     a+=5;
9     System.out.println("a = "+a);
10    a-=5;
11    System.out.println("a = "+a);
12    a*=5;
13    System.out.println("a = "+a);
14    a/=5;
15    System.out.println("a = "+a);
16    a%=5;
17    System.out.println("a = "+a);
18 }
```



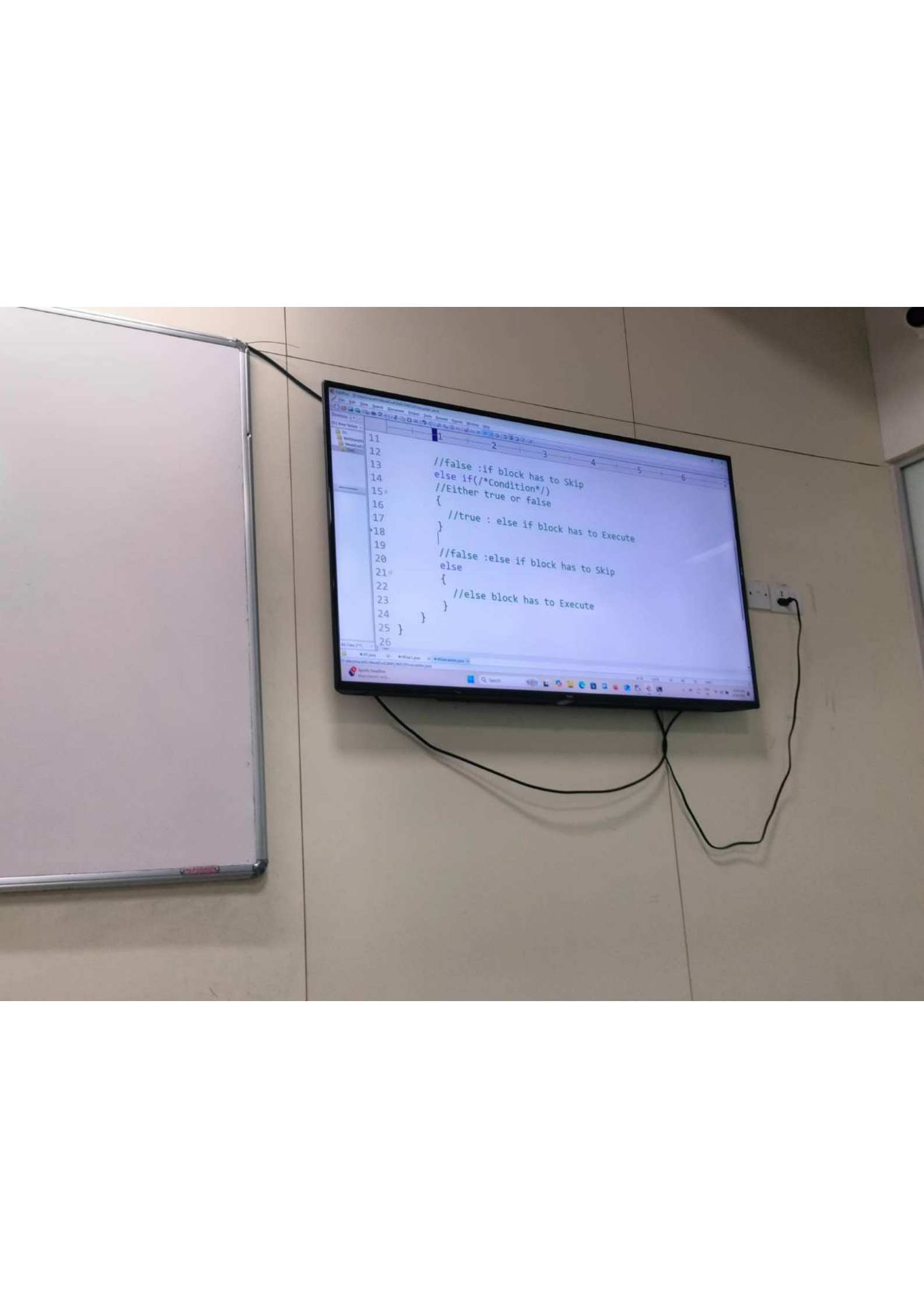
```
1 class P2
2 {
3
4     public static void main(String[] args)
5     {
6         char ch='$';
7         if ((ch>='A' && ch<='Z') || (ch>='a' && ch<='z'))
8         {
9             System.out.println("Alphabet");
10        }
11        else if ((ch>='0' && ch<='9'))
12        {
13            System.out.println("Digit");
14        }
15        else
16            System.out.println("Special Character");
17    }
18 }
```



```
1 class If1
2 {
3
4     public static void main(String[] args)
5     {
6         //Syntax:
7         if (*Condition*)
8         //Either true or false
9         {
10             //true : block has to Execute
11         }
12         //false : block has to Skip
13     }
14 }
```



```
4 {  
5  
6 //Syntax:  
7 if (/*Condition*/)  
8 //Either true or false  
9 {  
10 //true :if block has to Execute  
11 }  
12  
13 //false :if block has to Skip  
14 else if(/*Condition*/)  
15 //Either true or false  
16 {  
17 //true : else if block has to Execute  
18 }  
19 //false :else if block has to Skip
```



```
11  
12  
13 //false :if block has to Skip  
14 else if(*Condition*)  
15 //Either true or false  
16 {  
17     //true : else if block has to Execute  
18 }  
19  
20 //false :else if block has to Skip  
21 else  
22 {  
23     //else block has to Execute  
24 }  
25 }  
26
```


Operations:

→ Operands
↓
Value/Variable

→ Operator
↓
Symbol

Operands → Operators → Operands
Expression
we can perform

→ 3 types

* Unary op

* Binary op

* Ternary op

Unary op

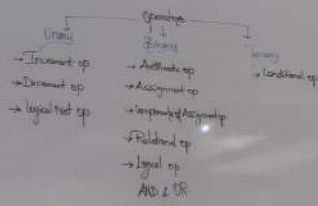
→ It accept only one value/variable and one operator

Binary op

→ It accept two value/variable and one operator

Ternary op

→ It will accept three value/variable and one operator



Relational op

- It is also called as Comparison op.
- It is an binary op.
- It gives return type value i.e. boolean type.
- It is comparing the operands.

Symbols: $>$, $>=$, $<$, $<=$, $!=$, $==$

Number

$>$
 $>=$
 $<$
 $<=$
 $!=$
 $==$

String & boolean
 $!=$
 $==$

Use

$op1 > op2$

$op1 >= op2$

$op1 < op2$

$op1 <= op2$

$op1 != op2$

$op1 == op2$

Description

L.H.S should be greater than R.H.S

L.H.S should be greater or equal to R.H.S

L.H.S should be less than R.H.S

L.H.S should be less or equal to R.H.S

L.H.S and R.H.S should not be same

L.H.S and R.H.S should be same

logical op

→ logical AND (&)

→ logical OR (||)

→ logical NOT (!)

→ Relational op is used to compare the values of the operands.

→ It gives boolean type value i.e. true or false.

→ It is comparing the condition.

Logical AND (&)	Logical OR ()	Logical NOT (!)
1 & 1	1 1	!1
1 & 0	1 0	!0
0 & 1	0 1	!1
0 & 0	0 0	!0

AND (&)

OR (||)

```
class L1 {  
    public static void main (String args[]) {  
        String shoe = "shoe";  
        double price = 100.0;  
        boolean aff = true;  
        System.out.println(shoe + price + aff);  
    }  
}
```

Relational op.

- It is also called as Comparison op.
- It is an binary op.
- It gives relation b/w value is boolean type.

Symbols: $>$, $>=$, $<$, $<=$, $!=$, $==$

Number

String & boolean

$!=$

$==$

Use

$op_1 > op_2$

$op_1 >= op_2$

$op_1 < op_2$

$op_1 <= op_2$

$op_1 != op_2$

$op_1 == op_2$

Description

L.H.S should be numeric R.H.S

L.H.S should be numeric R.H.S

L.H.S should be numeric R.H.S

L.H.S should be numeric R.H.S

L.H.S and R.H.S should be same

L.H.S and R.H.S should be same

Assignment

→ It is used to assign value

From R.H.S to L.H.S

Components of Assignment op

→ It is used to assign value

Arithmetic

Examples

Simple var

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$

$a = 5$



Meibharathi ravi

offline



These chats are monitored by admin

12/4/2020

```
import java.util.Scanner;
class W1
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter the Byte Value : ");
        byte b1=s.nextByte();
        short s1=b1;
        int i=b1;
        long l=b1;
        float f=b1;
        double d=b1;
        System.out.println(s1+"\n"+i+"\n"+l+"\n"+f+"\n"+d);
        s.close();
    }
}
```

QCR-SJFCJE-M8 01:11 PM

```
import java.util.Scanner;
class N1
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter the Double Value : ");
        double d=s.nextDouble();
        float f=(float)d;
        long l=(long)d;
        int i=(int)d;
        short s1=(short)d;
    }
}
```

Message...



offline

These chats are monitored by admin

```
s.close();
```

```
}
```

```
}
```

12/4/2025

QCR-SJFCJE-M8 01:11 PM

```
import java.util.Scanner;
class N1
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter the Double Value : ");
        double d=s.nextDouble();
        float f=(float)d;
        long l=(long)d;
        int i=(int)d;
        short s1=(short)d;
        byte b=(byte)d;
        System.out.println(f+"\n"+l+"\n"+i+"\n"+s1+"\n"+b);
        s.close();
    }
}
```

QCR-SJFCJE-M8 01:11 PM

Assignment:

-->WAP to convert value into char datatype literals.

-->Execute remaining datatypes in widening and narrowing

QCR-SJFCJE-M8 01:12 PM

op1 + op2

op1 - op2

* op2

/ op2

/ op2

op1	op2	Result
byte	byte	Error
short	short	Error
char	char	Error
int	int	Success
long	long	Success
float	float	Success
double	double	Success
boolean	boolean	Error
String	String	Success (Concat)

Assignment

⇒ If typed to assign value from R.H.S to L.H.S
 (operator of Assignment op)
 → It is combination of Arithmetic & Assignment

Arithmetic	Assignment
+	→ = → + =
-	→ = → - =
*	→ = → * =
/	→ = → / =
%	→ = → % =

Developer View	Compiler View
a + = 5	$a = a + 5$
a - = 5	$a = a - 5$
a * = 5	$a = a * 5$
a / = 5	$a = a / 5$
a % = 5	$a = a \% 5$

```

1 public static void main(String[] args) {
2     // ...
3     int a = 10;
4     // ...
5     a += 5;
6     System.out.println("a = " + a);
7     a -= 5;
8     System.out.println("a = " + a);
9     a *= 5;
10    System.out.println("a = " + a);
11    a /= 5;
12    System.out.println("a = " + a);
13    a %= 5;
14    System.out.println("a = " + a);
15 }
  
```


Operators	Associativity	Type
++ --	Right to left	Unary postfix
++ -- + - ~ (type)	Right to left	Unary prefix
* / %	Left to right	Multiplicative
+ -	Left to right	Additive
<< >> >>>	Left to right	Shift
< <= > >=	Left to right	Relational
== !=	Left to right	Equality
&	Left to right	Boolean Logical AND
^	Left to right	Boolean Logical Exclusive OR
	Left to right	Boolean Logical Inclusive OR
&&	Left to right	Conditional AND
	Left to right	Conditional OR
?:	Right to left	Conditional
= += -= *= /= %=	Right to left	Assignment



Q Search



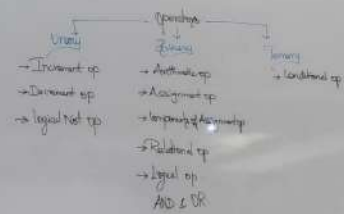
Arithmetic op

→ Numeric calculations

- $+$ → add the value (or) merge the value (concat) → $ep1 + ep2$
- $-$ → Subtract the value by $ep2$ to $ep1$ → $ep1 - ep2$
- $*$ → Multiply the value from $ep1$ to $ep2$ → $ep1 * ep2$
- $/$ → If provides Quotient → $ep1 / ep2$
- $\%$ → If gives Remainder → $ep1 \% ep2$

op Addition

ep1	ep2	Result
byte	byte	Error
short	short	Error
char	char	Error
int	int	Success
long	long	Success
float	float	Success
double	double	Success
boolean	boolean	Error
String	String	Success (concat)



Operators

→ Operands
↓
Value/variable
→ Operator
↓
Symbol

Operands → Operators → operands
By whom we can perform

→ Types

- * Unary op
- * Binary op
- * Ternary op

Unary op

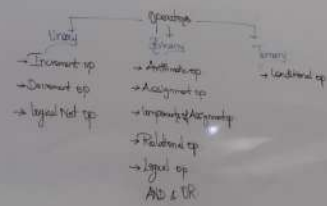
→ It accept only one value/variable and one operator

Binary op

→ It accept two value/variable and one operator

Ternary op

→ It will accept three value/variable and one operator



→ Step 1: declare obj ref for
Scanner obj ref name = new Scanner

Operators
In Java, operators are the
symbols.

classified into three
Unary, Binary, Ternary.

types of operators
Arithmetic operator
Assignment operator
Comparison operator
Logical operator
Increment/Decrement
Conditional operator

Conditional op.

→ It is also called as Ternary op.

→ ? :

→ ? → Decision Pending.

→ : → Decide either one.

Syntax:

Op1 ? Op2 : Op3 ;
→ (Condition)
true ———→
false ———→

→ It depends on op1 and op2 data or literals.

→ WAP to check the lar var of three numbers?

→ WAP to check whether it is leap year or not?