Practice:

- Accept two integers and perform arithmetic operations.
- Accept a character and check character is vowel or not.
- Accept a amount and find number of notes required for a given amount of money.

Practice:

- Accept two integers and perform arithmetic operations.
- Accept a character and check character is vowel or not.
- Accept a amount and find number of notes required for a given amount of money.

Rules for switch case:

- Switch is taking the argument, the allowed arguments are
  byte, short, int, char (primitive data types)
  Byte, Short, Integer, Character(wrapper classes)
  String

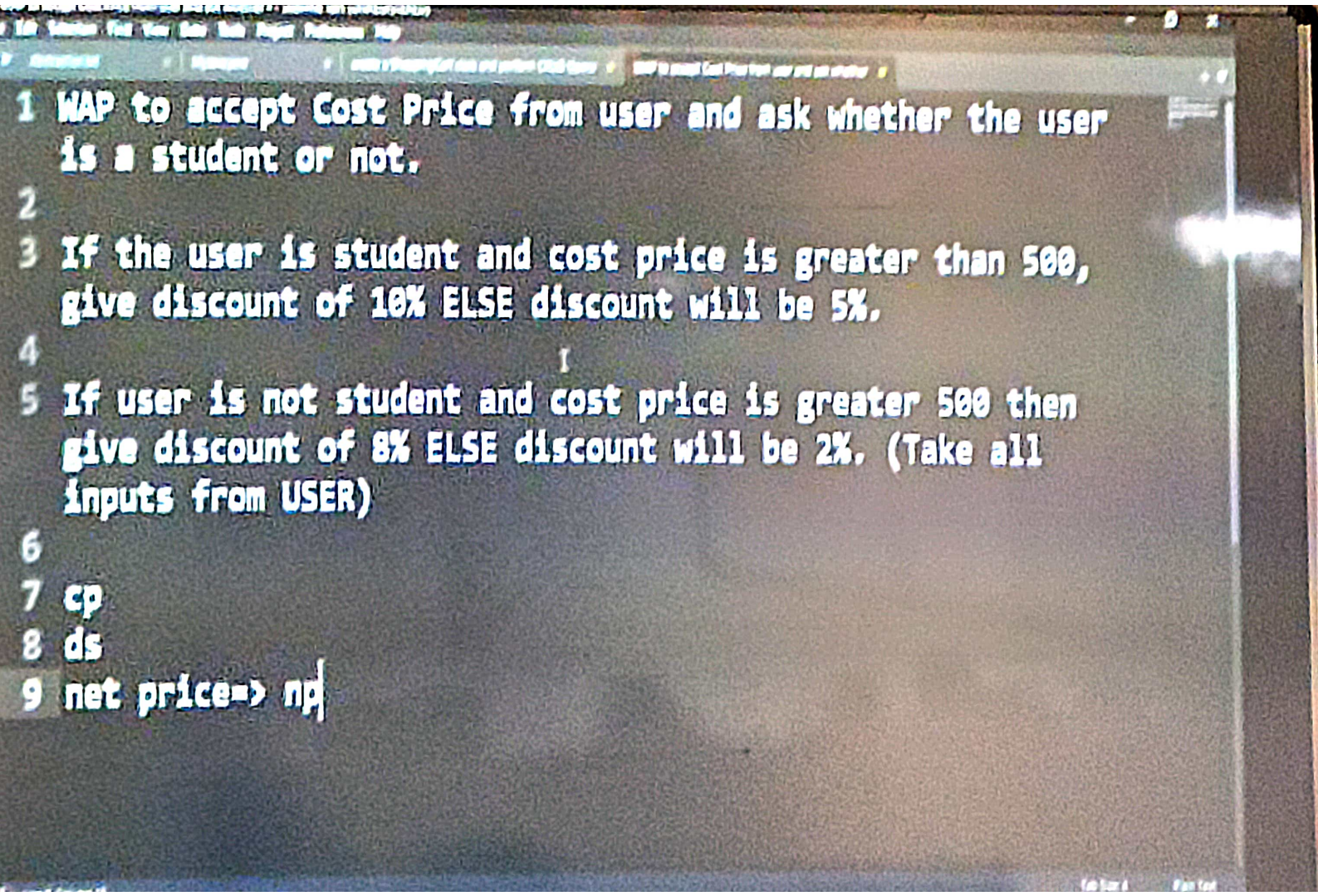
- Inside the switch it is possible to declare more than one case but it is possible to declare only one default statement
- float, double, long are not allowed as a switch argument because these are having too large values
- Inside the switch statement **break, default** and **case** are optional
- Inside the switch the case labels must be unique

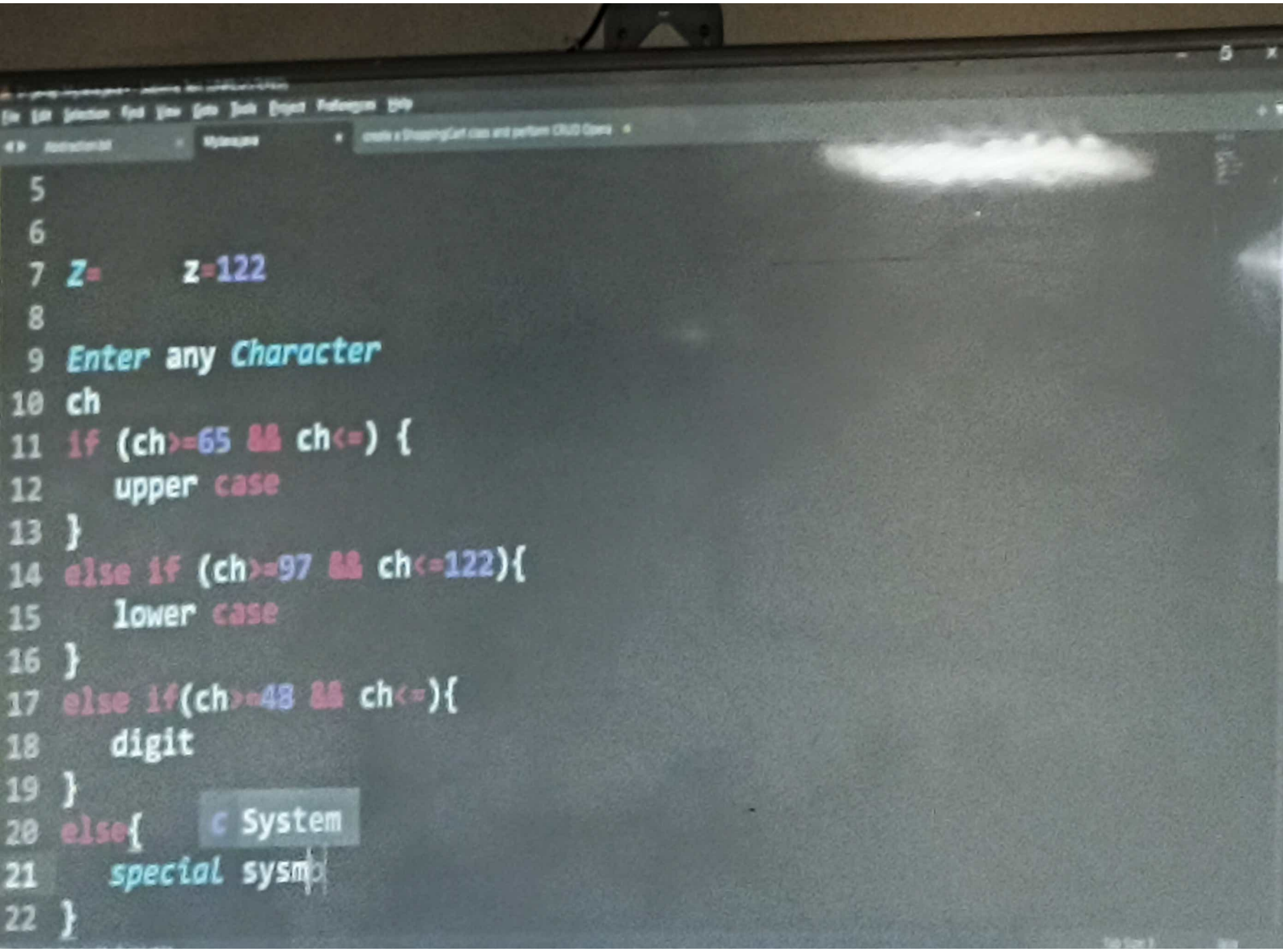
Switch case It is the multiple choice statement and a substitute for if else ladder

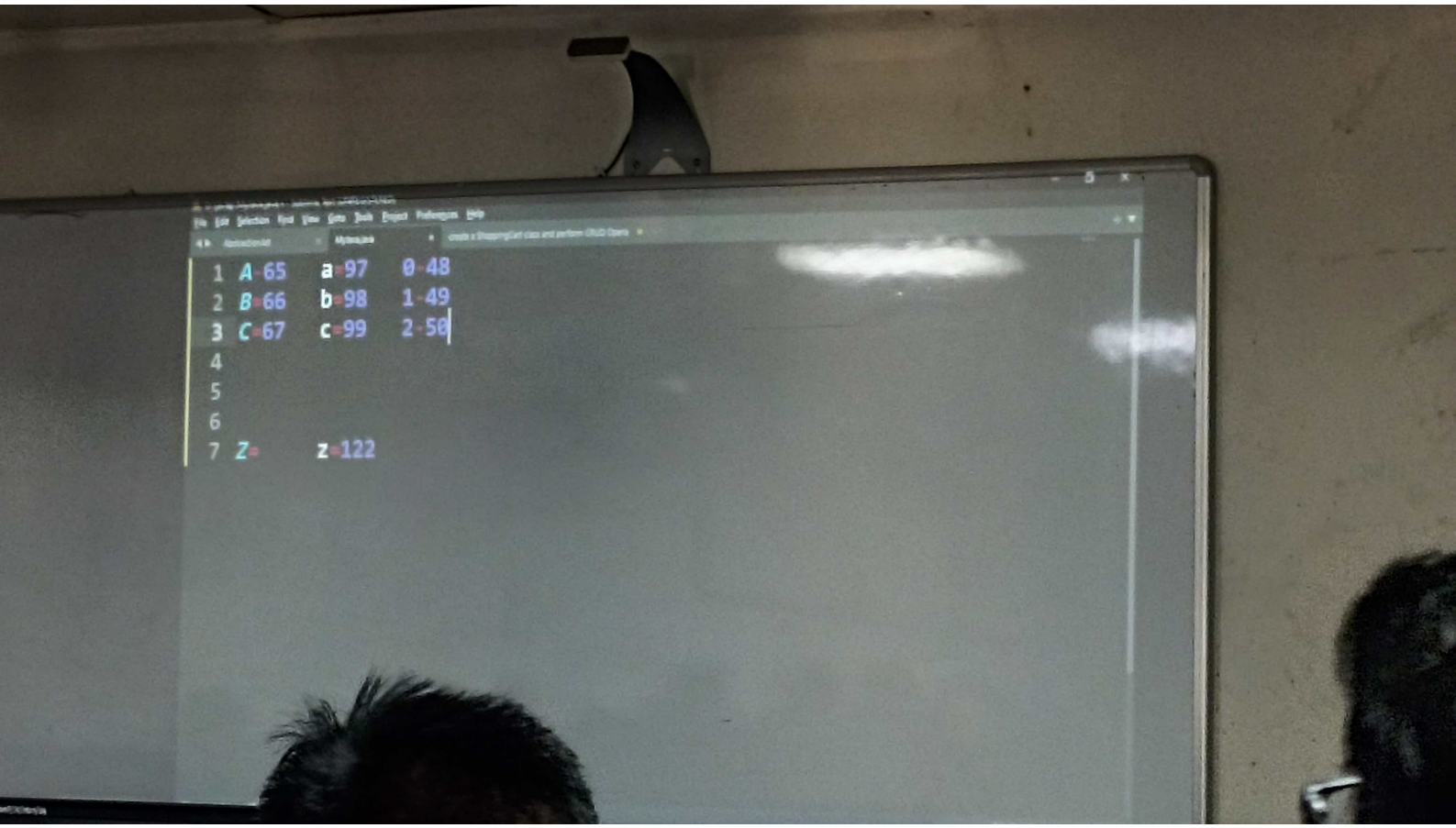
Syntax:

```
switch (n) {
    case 1: // code to be executed if n = 1;
        break;
    case 2: // code to be executed if n = 2;
        break;
    case 3: // code to be executed if n = 3;
        break;

    default: // executed if n doesn't match any cases
}
```


1 WAP to accept Cost Price from user and ask whether the user is a student or not.
2
3 If the user is student and cost price is greater than 500, give discount of 10% ELSE discount will be 5%.
4
5 If user is not student and cost price is greater 500 then give discount of 8% ELSE discount will be 2%. (Take all inputs from USER)
6
7 cp
8 ds
9 net price=> np


5
6
Z= z=122
8
Enter any Character
ch
if (ch>=65 && ch<=) {
upper case
}
else if (ch>=97 && ch<=122){
lower case
}
else if(ch>=48 && ch<=){
digit
}
else{
System
special sysm
}


1 A=65 a=97 0=48
2 B=66 b=98 1=49
3 C=67 c=99 2=50
7 Z= z=122

Practice:

- Accept 4 paper marks find total, percentage and assign a grade. The method of assigning grade is

  If percentage >= 90% : Grac A
  If percentage >= 80% : Grade B
  If percentage >= 70% : Grade C
  If percentage >= 60% : Grade D
  If percentage >= 40% : Grade E
  If percentage < 40% : fail

- Accept a character and check character is upper case, lower case, digit or special symbol

Practice:

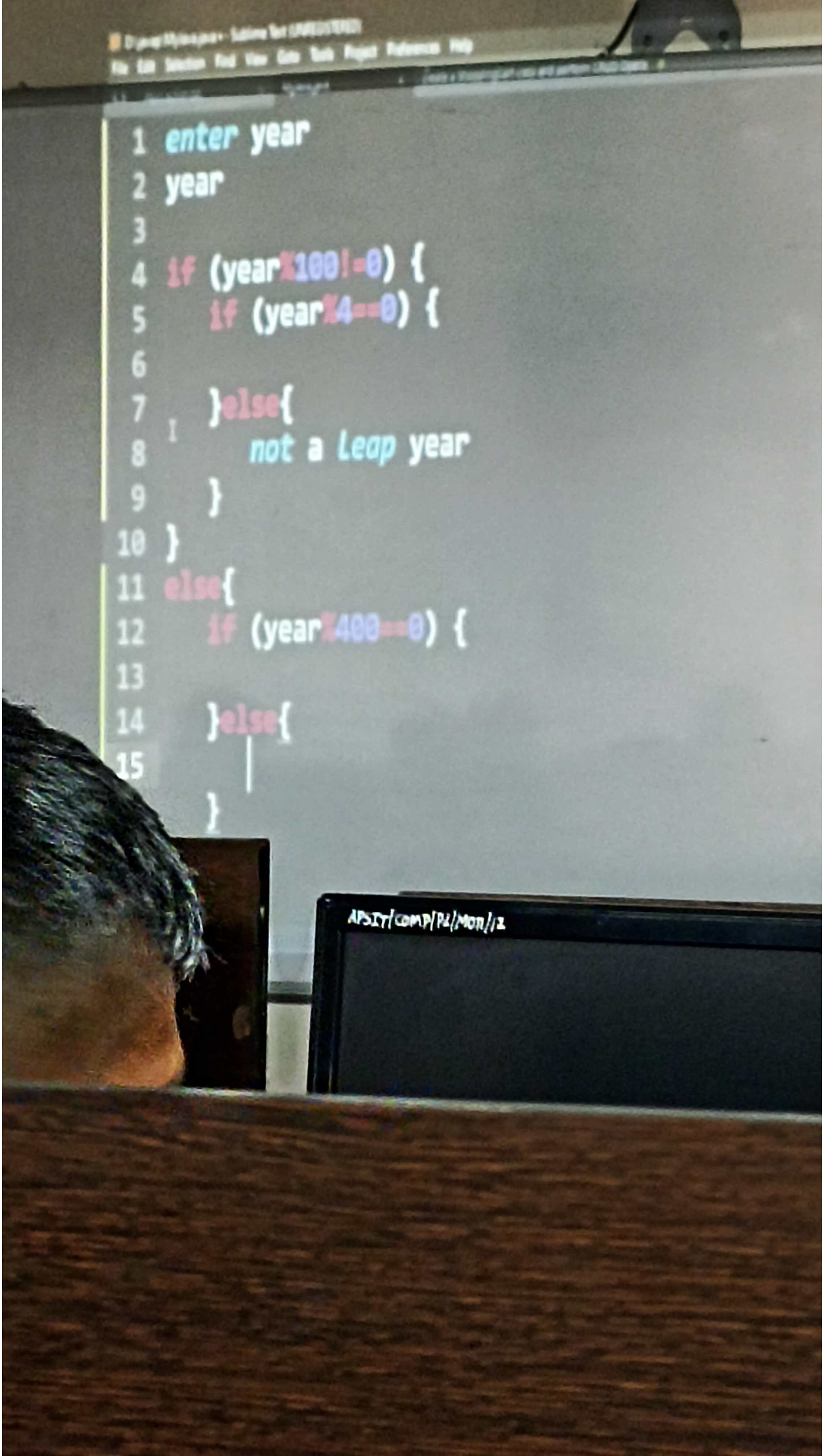
- Accept 4 paper marks find total, percentage and assign a grade. The method of assigning grade is

  If percentage >= 90% : Grade A
  If percentage >= 80% : Grade B
  If percentage >= 70% : Grade C
  If percentage >= 60% : Grade D
  If percentage >= 40% : Grade E
  If percentage < 40% : fail

- Accept a character and check character is upper case, lower case, digit or special symbol

Find the grade of a student when 4 subject marks are given.

```
class Pro  {
  public static void main(String args[])      {
        double m1=60,m2=75,m3=66,m4=84,total,per;
    char grade=' ';
    total = m1+m2+m3+m4;            per = total/4.0;
    if(per>=85)
      grade='A';
    else if(per>=75 && per<85)
      grade='B';
    else if(per>=55 && per<75)
      grade='C';
    else if(per>=40 && per<55)
      grade='D';
    else
      grade='E';
   System.out.println("percentage "+per+" Grade is "+grade);
  } }
```


1 enter year
2 year
3
4 if (year%100!=0) {
5 if (year%4==0) {
6
7 }else{
8 not a Leap year
9 }
10 }
11 else{
12 if (year%400==0) {
13
14 }else{
15
}
APSIT/COMP/PL/MOM/12

Practice:

- Accept 3 numbers and find largest number.
- Accept a year and check leap year or not (use century and non century concept).

Program to check largest number among three number

```
class Pro
{
   public static void main(String args[])     {
      int a=3,b=7,c=22,big;
      if(a>b)       {
                if(a>c)
                   big=a;
                else
                   big=c;
      }
      else       {
            if(b>c)
               big=b;
            else
               big=c;
      }
       System.out.println("Big value is "+big);
```

Practice:

- Accept 3 paper marks, if paper mark less than 45 show "fail" or show "pass".
- Accept 3 paper marks, find total and percentage, if percentage >= 62 and gender is female, she can take admission else can't take admission.
- Accept day and check day is weekend day or working day.
- Accept a character and check character is vowel or not.

6. Division of two numbers

7. Find last digit of number

8. Add two digits of a number

9. Sum of 3 digits of a number

10. Sum of 4 digits of a number

11. Reverse of a 4 digit number

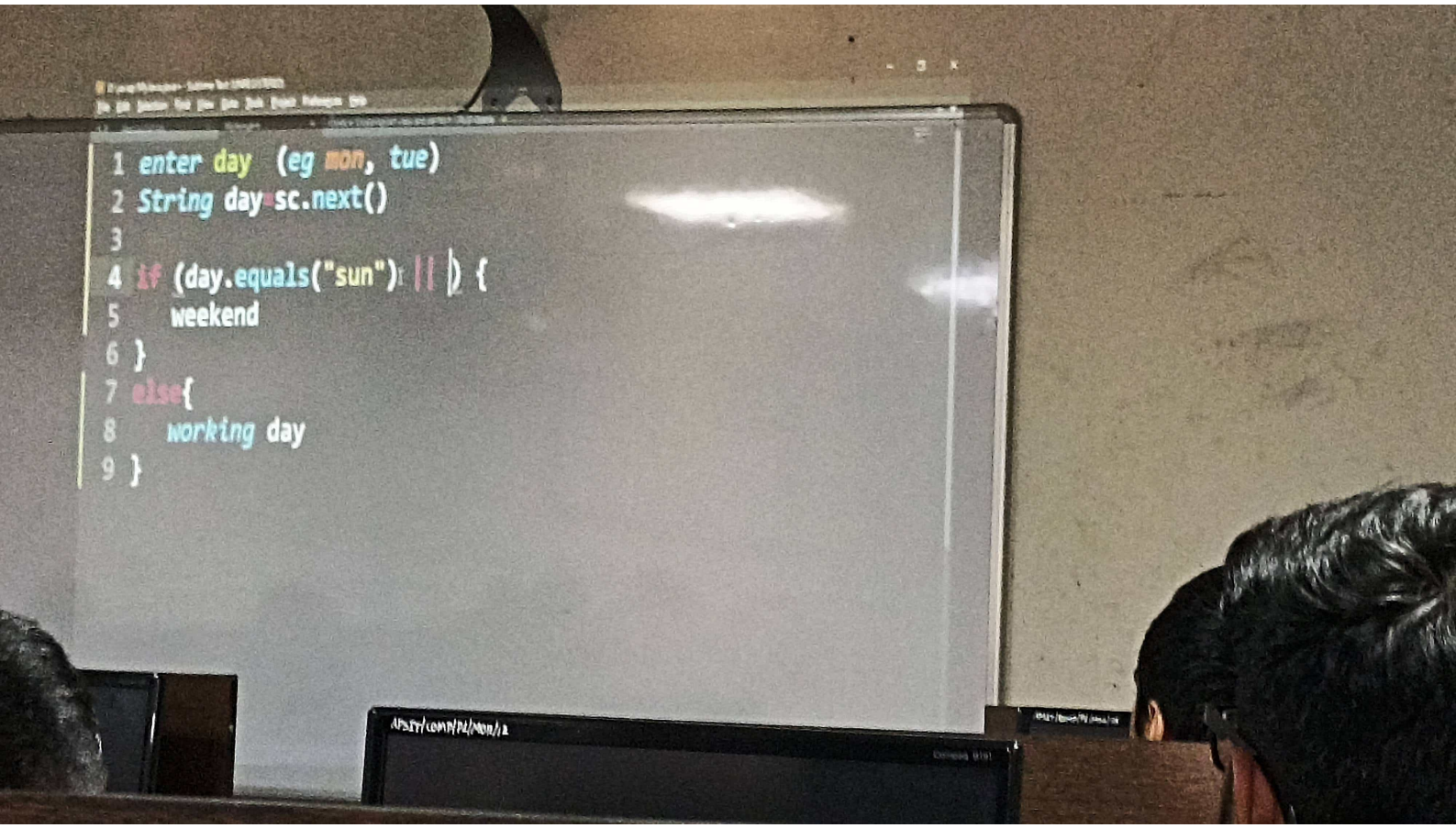
12. Swap using third variable

13. Swap without third variable

14. Average of three numbers

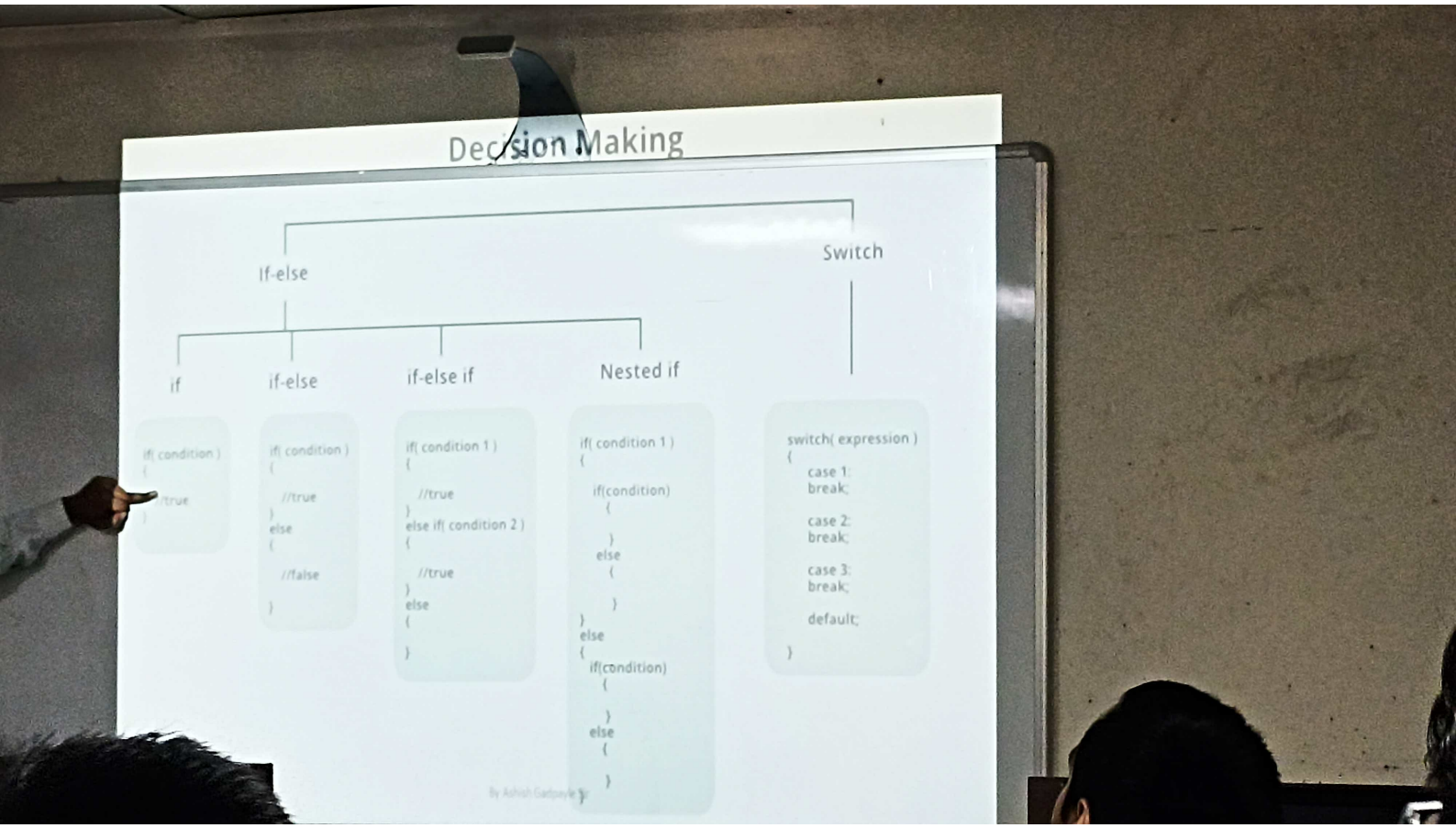
Program to check whether the entered number is positive negative or zero

```
class IfPro{
   public static void main(String args[])
   {
      int n=0;
   if(n>0)
      System.out.println("Number is Positive");
   if(n<0)
      System.out.println("Number is Negative");
   if(n==0)
      System.out.println("Number is zero");
   }
```


enter day (eg mon, tue)
String day=sc.next()
if (day.equals("sun") ||) {
weekend
}
else{
working day
}

Practice:

- Accept 3 paper marks, if paper mark less than 45 show "fail" or show "pass".
- Accept 3 paper marks, find total and percentage. if percentage >= 62 and gender is female, she can take admission else can't take admission.
- Accept day and check day is weekend day or working day.
- Accept a character and check character is vowel or not.

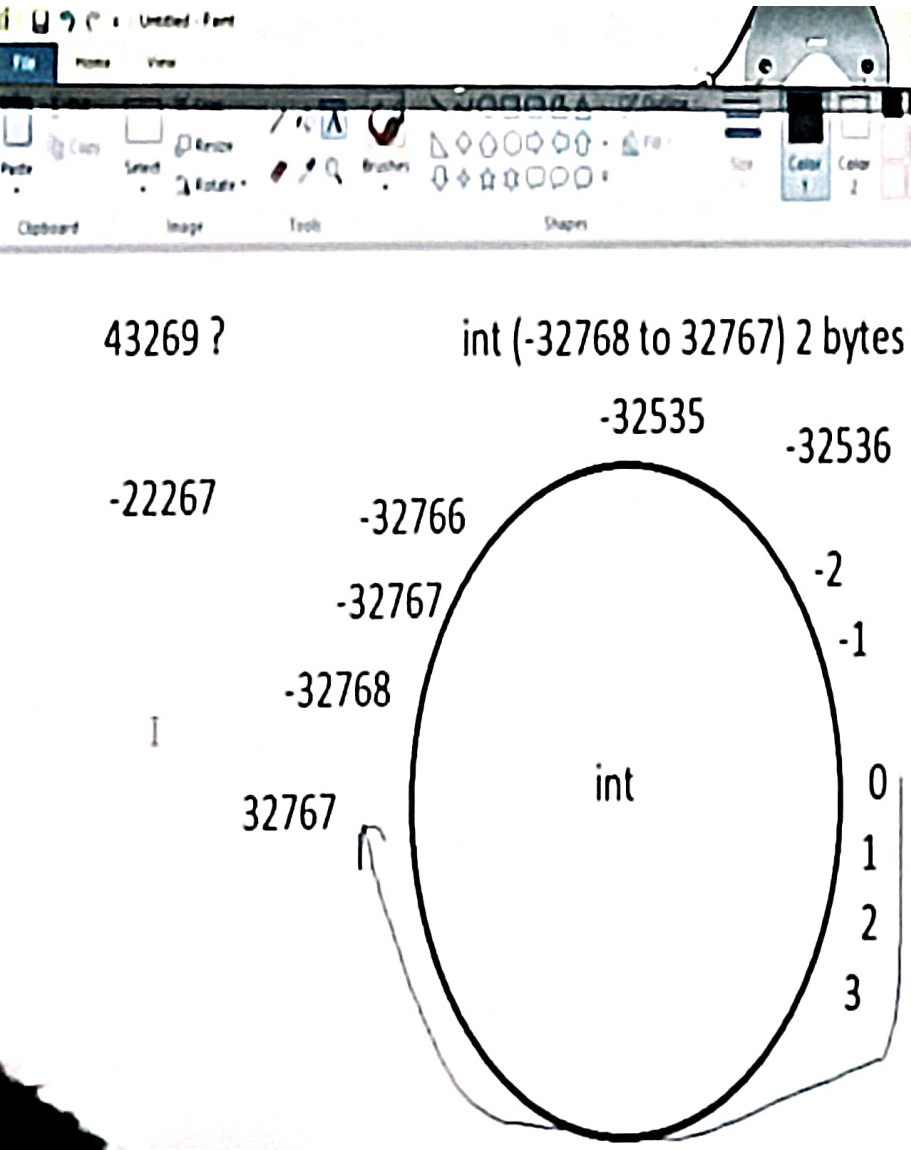

Decision Making
If-else
Switch
if
if-else
if-else if
Nested if
if(condition)
{
//true
}
if(condition)
{
//true
}
else
{
//false
}
if(condition 1)
{
//true
}
else if(condition 2)
{
//true
}
else
{
}
if(condition 1)
{
if(condition)
{
}
else
{
}
}
else
{
if(condition)
{
}
else
{
}
}
switch(expression)
{
case 1:
break;
case 2:
break;
case 3:
break;
default;
}

Decision making statements available in Java are:

- if statement
- if..else statements
- nested if statements
- if-else-if ladder
- switch statements

There come situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code.

Selection statement/Decision making statements in programming languages decides the direction of flow of program execution.


43269 ?
int (-32768 to 32767) 2 bytes
-32535
-32536
-22267
-32766
-2
-32767
-1
-32768
int
32767
0
1
2
3
int i=33000;
33000
32767
233
32768
233
32535

6. Division of two numbers

7. Find last digit of number

8. Add two digits of a number

9. Sum of 3 digits of a number

10. Sum of 4 digits of a number

11. Reverse of a 4 digit number

12. Swap using third variable

13. Swap without third variable

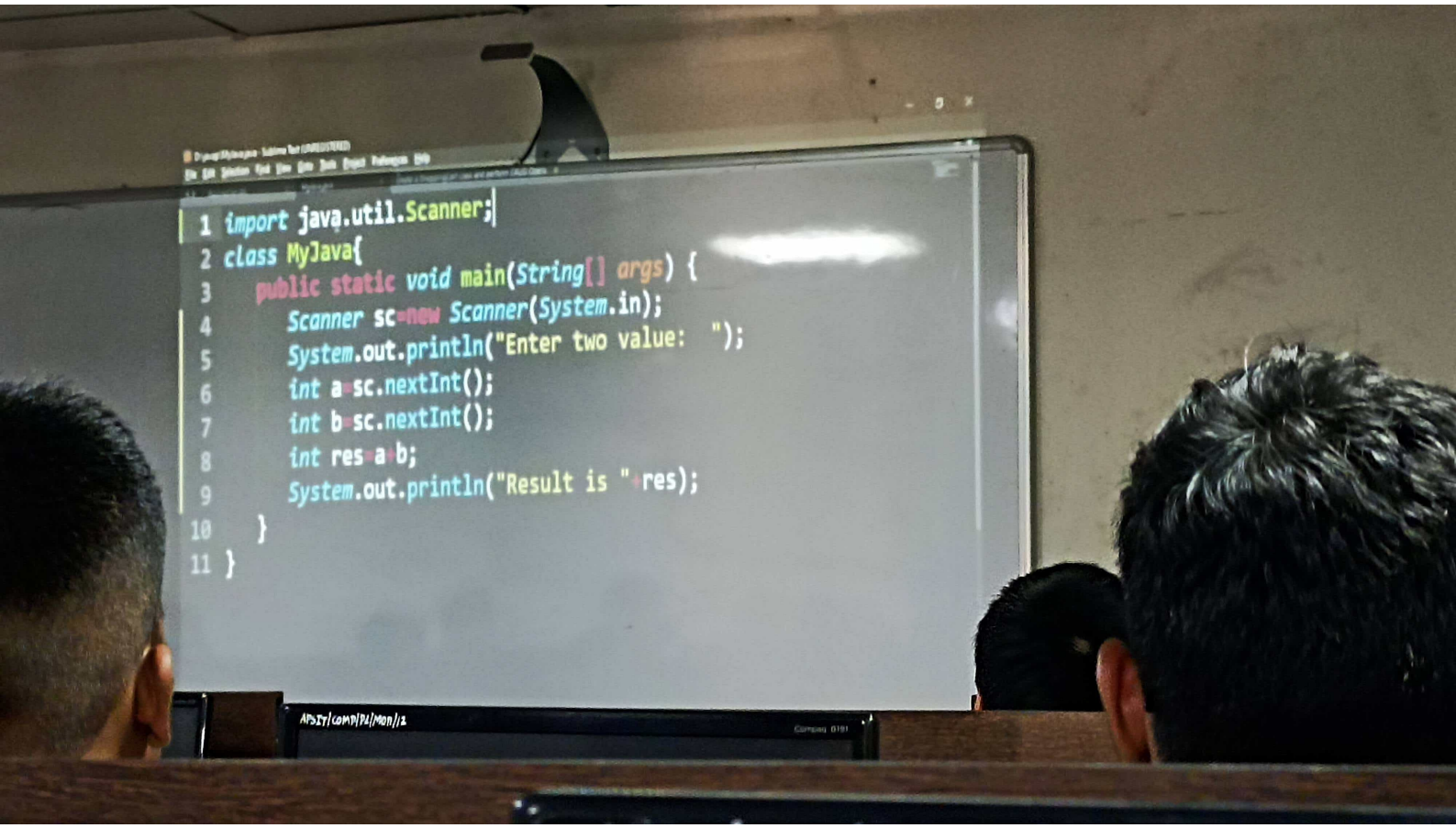
14. Average of three numbers

```
import java.util.Scanner;
class ScannerDemo  {
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
         String name = sc.nextLine();          // String input
         char gender = sc.next().charAt(0);    // Character input
         int age = sc.nextInt();               // Numerical data input
      long mobileNo = sc.nextLong();           // Long input
      double cgpa = sc.nextDouble();           // Double input
      // Print the values
      System.out.println("Name:"+name);
      System.out.println("Gender:"+gender);
      System.out.println("Age:"+age);
      System.out.println("Mobile Number:"+mobileNo);
      System.out.println("CGPA:"+cgpa);
```

- Scanner class is availa in **java.util** package used to take input of the primitive types like byte, **short, int, long, float, double, boolean** etc.
- It is the easiest way to read input in a java program.
- To create an object of Scanner class, we usually pass the predefined object ystem.in
- System.in represents the standard input stream.
- To read numerical values of a certain data type **XYZ**, the function to use is **nextXYZ().**

  For example, to read a value of type **short**, we can use **nextShort()**

```
import java.util.Scanner;
class ScannerDemo   {
  public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
       String name = sc.nextLine();            // String input
       char gender = sc.next().charAt(0);      // Character input
       int age = sc.nextInt();                 // Numerical data input
    long mobileNo = sc.nextLong();             // Long input
    double cgpa = sc.nextDouble();             // Double input
    // Print the values
    System.out.println("Name:"+name);
    System.out.println("Gender:"+gender);
    System.out.println("Age:"+age);
    System.out.println("Mobile Number: "+mobileNo);
    System.out.println("CGPA:"+cgpa);
  }
```


import java.util.Scanner;
class MyJava{
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.println("Enter two value: ");
int a=sc.nextInt();
int b=sc.nextInt();
int res=a+b;
System.out.println("Result is "+res);
}
}
APSIT/COMP/BE/MON//2

- Scanner class is available in **java.util** package used to take input of the primitive types like **byte, short, int, long, float, double, boolean** etc.
- It is the easiest way to read input in a java program.
- To create an object of Scanner class, we usually pass the predefined object System.in
- System.in represents the standard input stream.
- To read numerical values of a certain data type **XYZ**, the function to use is **nextXYZ()**.

  For example, to read a value of type **short**, we can use **nextShort()**

What is the Output?

```
class EX11
{
	public static void main(String args[])
	{
		float f=0.7f;
		if (f==0.7)
			System.out.println("Hello");
		else
			System.out.println("Hi");
	}
}
```

Output

Hi

```
class E12
{
public static void main(String args[])
{
 float a=3.14,
 System.out.println(a);
}
}
```

```
class E11
{
        public static void main(String args[])
        {
         boolean b=true;
        System.out.println(b);
        System.out.println(!b);
        b=!b;
        System.out.println(b);
        System.out.println(!b);
        }
}
```

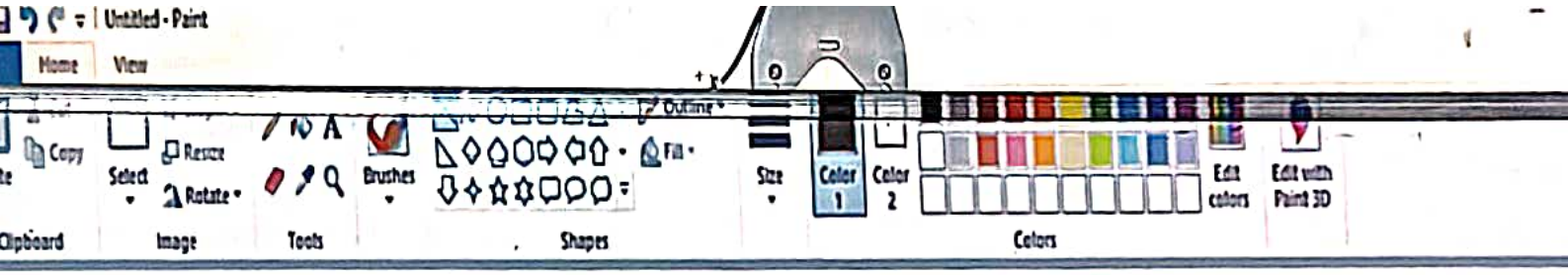
Outp

true
false
false
true

```
class E16
{
public static void main(String args[])
{
 char c=65;
 char ch='A';
 System.out.println(c);
 System.out.println(ch);
}
}
```

Output

```
A
A
```



internal

int 4 byte

float a=

4 byte

double 8 bytes

3.14

```
class E21
{
        public static void main(String args[])
        {
         int i=65;
         System.out.println(i);
         System.out.println((char)i);
        }
}
```

Output

```
65
A
```

```
class E22
{
	public static void main(String args[])
	{
	 byte b=9561467;
	 System.out.println(b);
	}
}
```

Output

```
E22.java:5: error: incompatible types: possible lossy conversion from int to
byte
	byte b=9561467;
	     ^
1 error
```

What is the Output?

```
class E22
{
	public static void main(String args[])
	{
	 byte b=9561467;
	 System.out.println(b);
	}
}
```

Output

```
E22.java:5: error: incompatible types: possible lossy conversion from int to
byte
	byte b=9561467;
	     ^
1 error
```

What is the Output?

```
class E21
{
        public static void main(String args[])
        {
         int i=65;
         System.out.println(i);
         System.out.println((char)i);
        }
```

Output

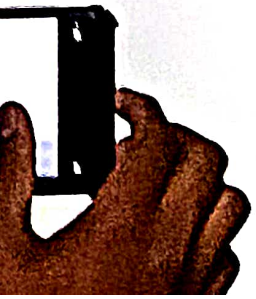
```
65
A
```

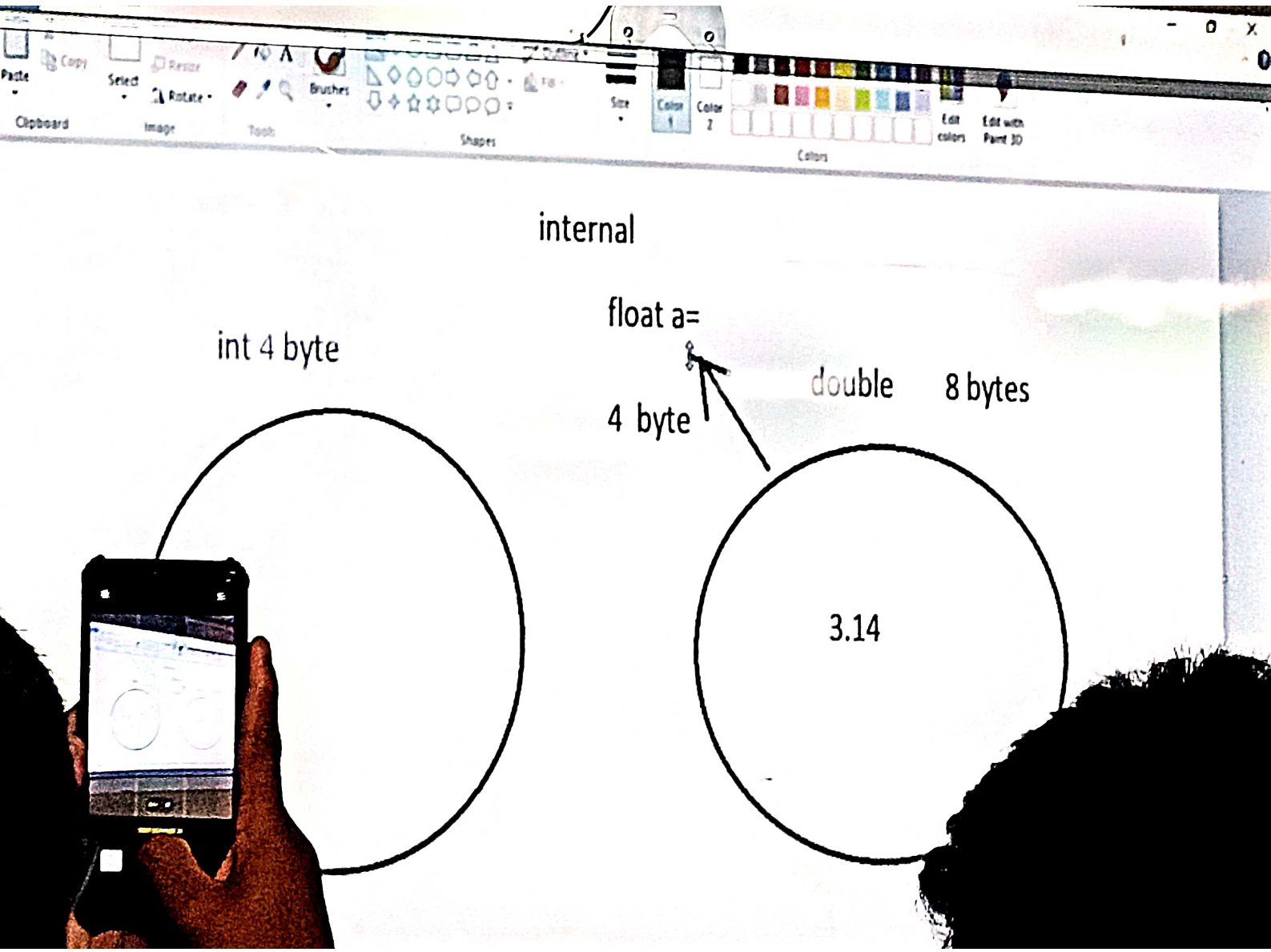
What is the Output?

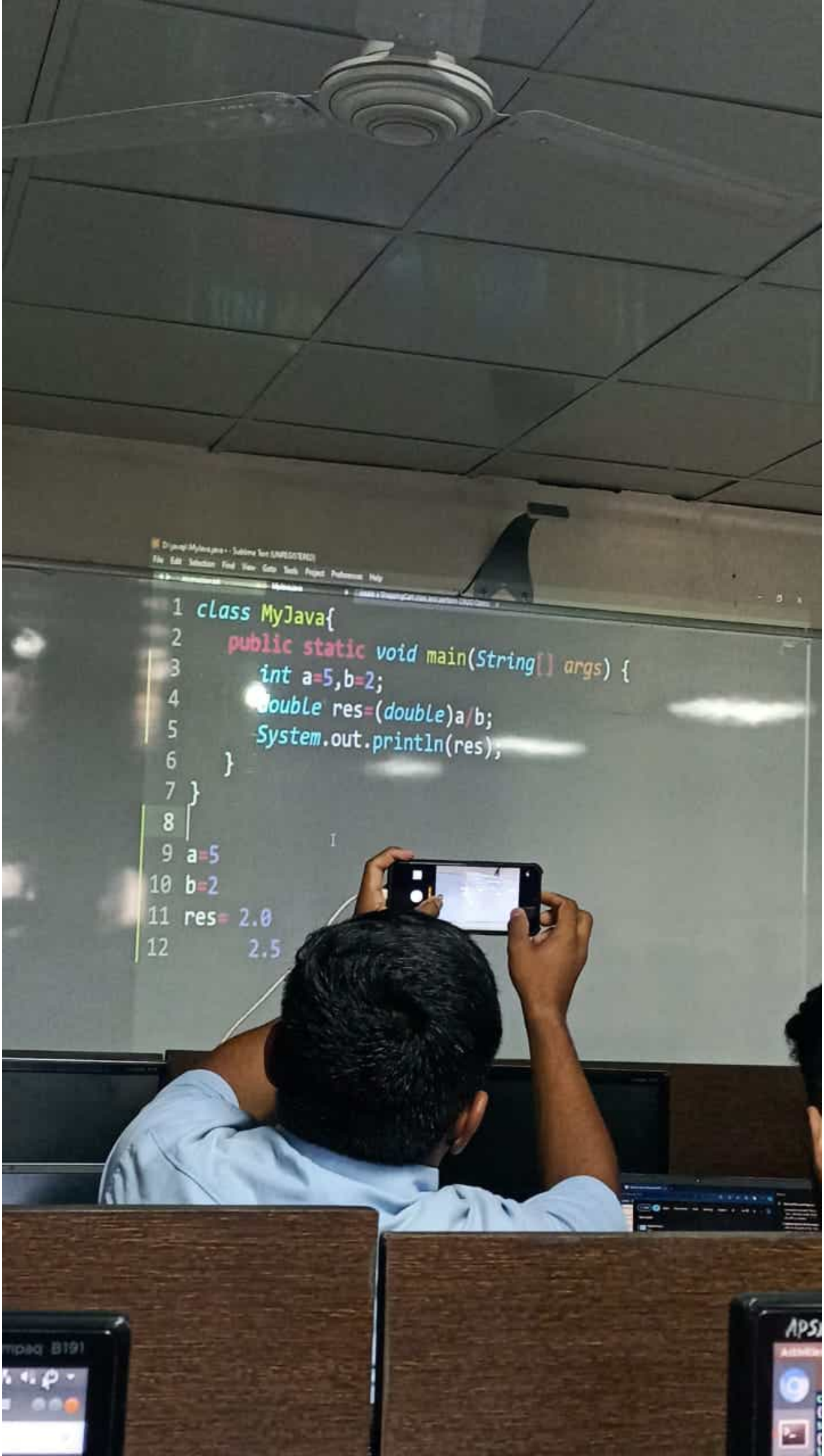
```
class EX11
{
    public static v     main(String args[])
    {
        float f=0.7f;
        if (f==0.7)
            System.out.println("Hello");
        else
            System.out.println("Hi");
    }
```

Output

Hi




Paste
Copy
Select
Rotate
Brushes
Clipboard
Image
Tools
Shapes
Size
Color 1
Color 2
Colors
Edit colors
Edit with Paint 3D
internal
int 4 byte
float a=
4 byte
double
8 bytes
3.14


class MyJava{
public static void main(String[] args) {
int a=5,b=2;
double res=(double)a/b;
System.out.println(res);
}
}
a=5
b=2
res= 2.0
2.5

The process of converting data one type to another type is called type casting

There are two types of type casting

- **Implicit type casting /widening/up casting**

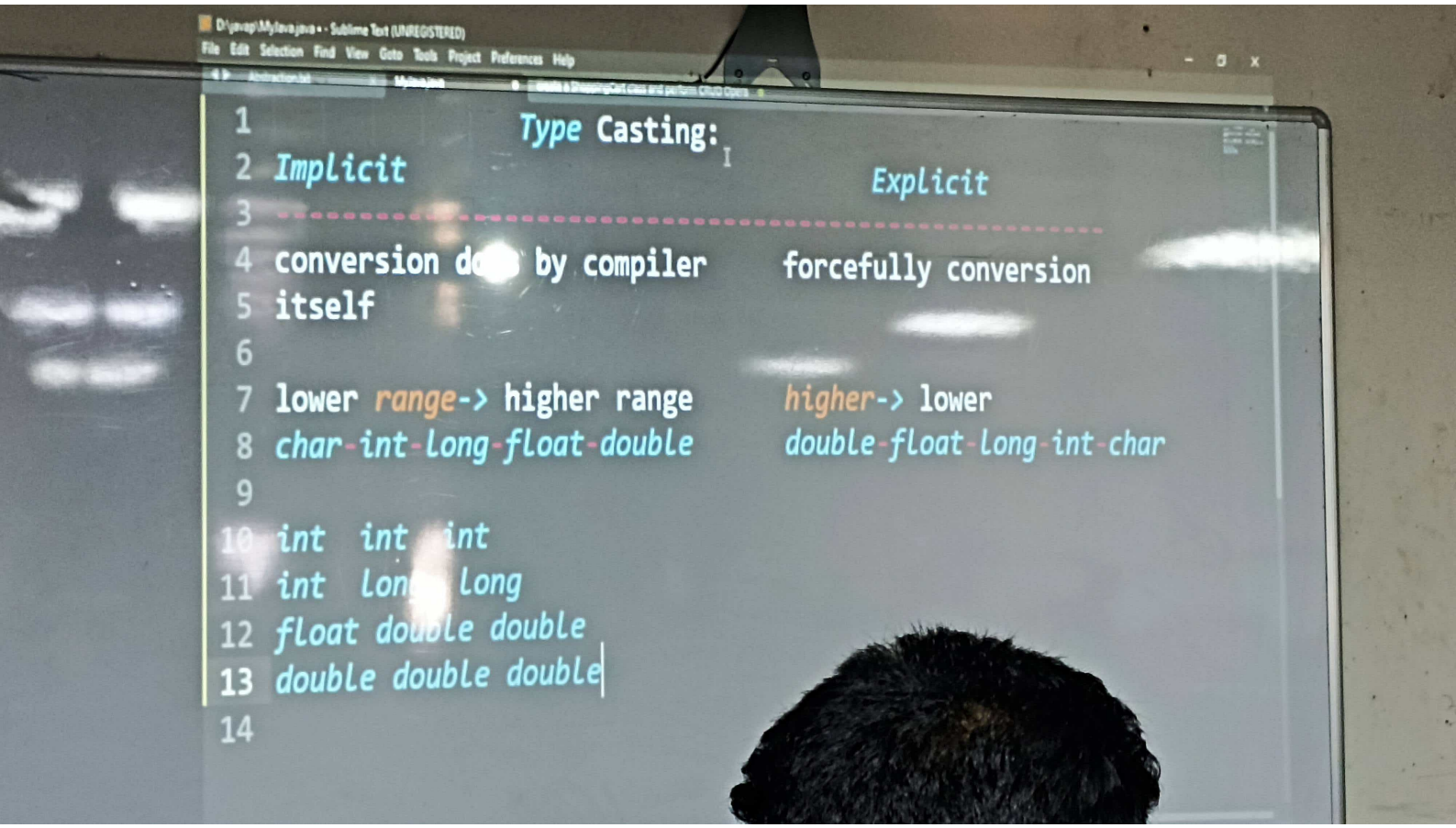
  This involves the conversion of a smaller data type to the larger type size.

  byte -> short -> char -> int -> long -> float -> double

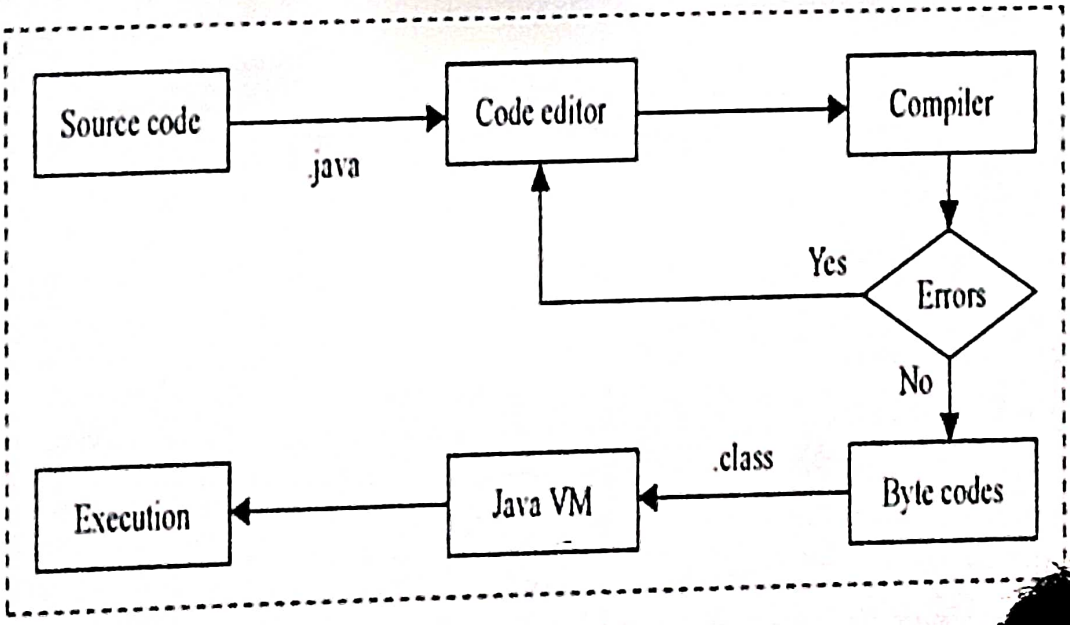
- **Explicit type casting /narrowing/down casting**

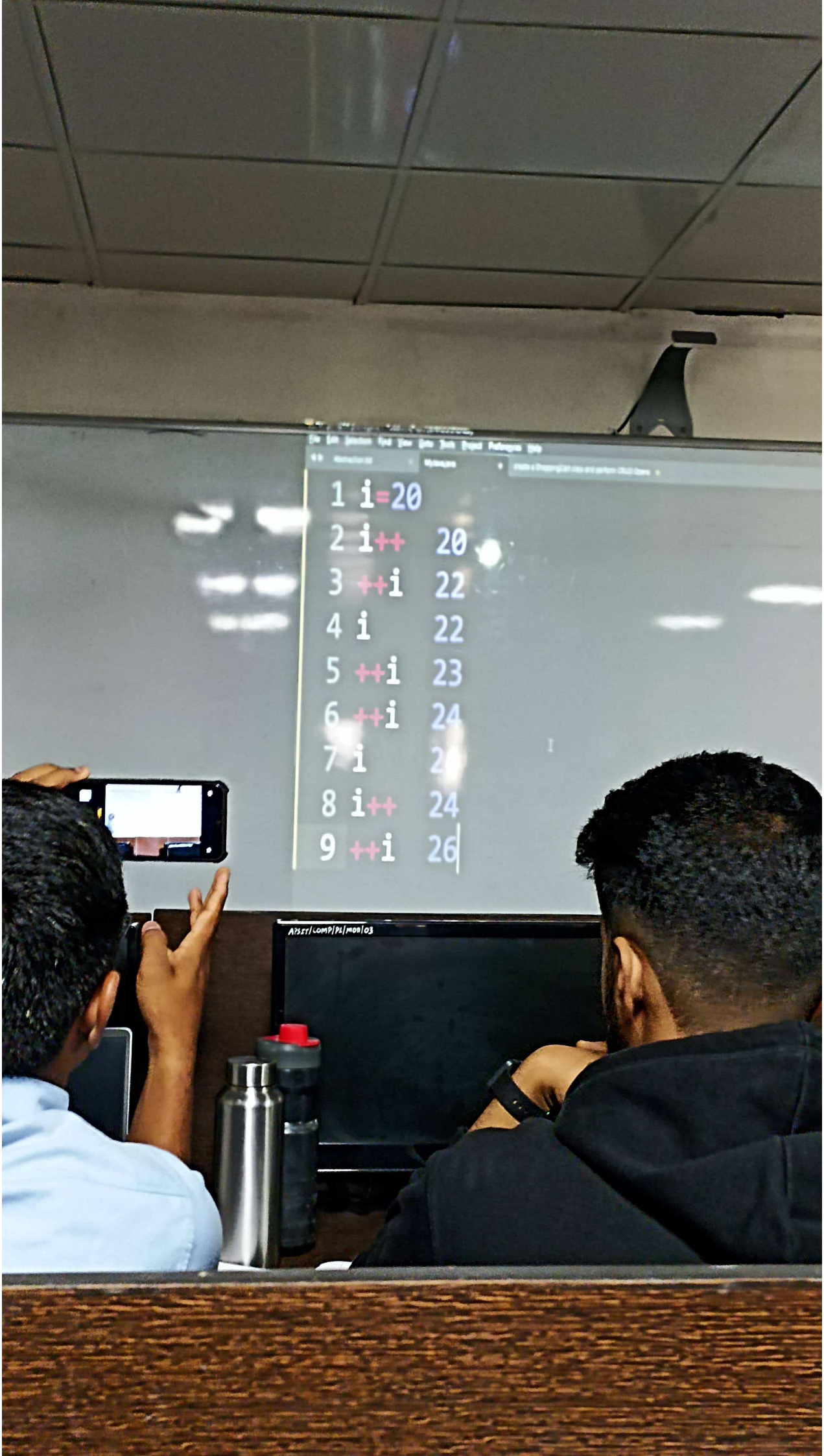
  This involves converting a larger data type to a smaller size type

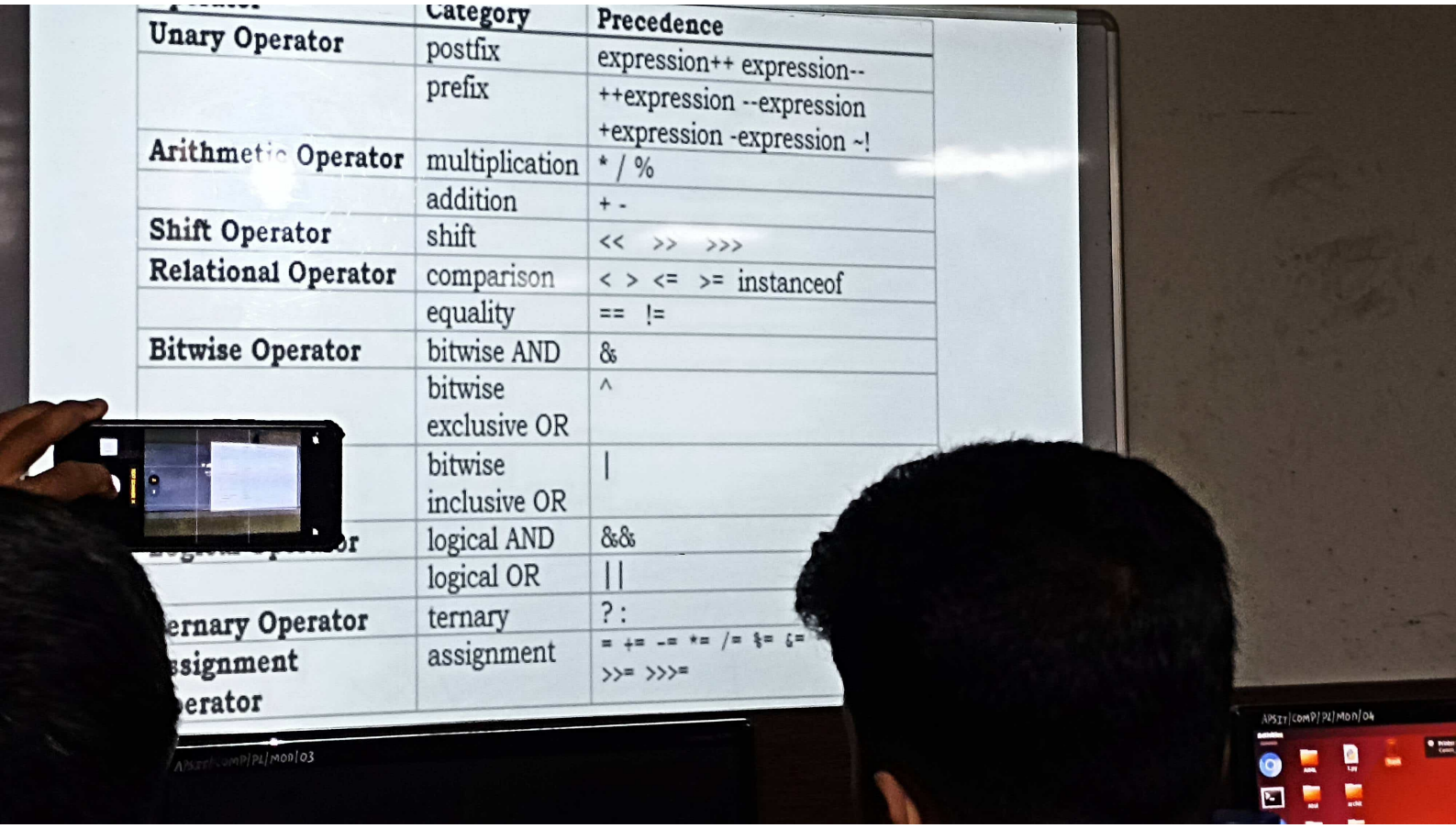
  double -> float -> long -> int -> char -> short -> byte


Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Type Casting:
Implicit
Explicit
conversion do by compiler
itself
forcefully conversion
lower range-> higher range
char-int-long-float-double
higher-> lower
double-float-long-int-char
int int int
int long long
float double double
double double double

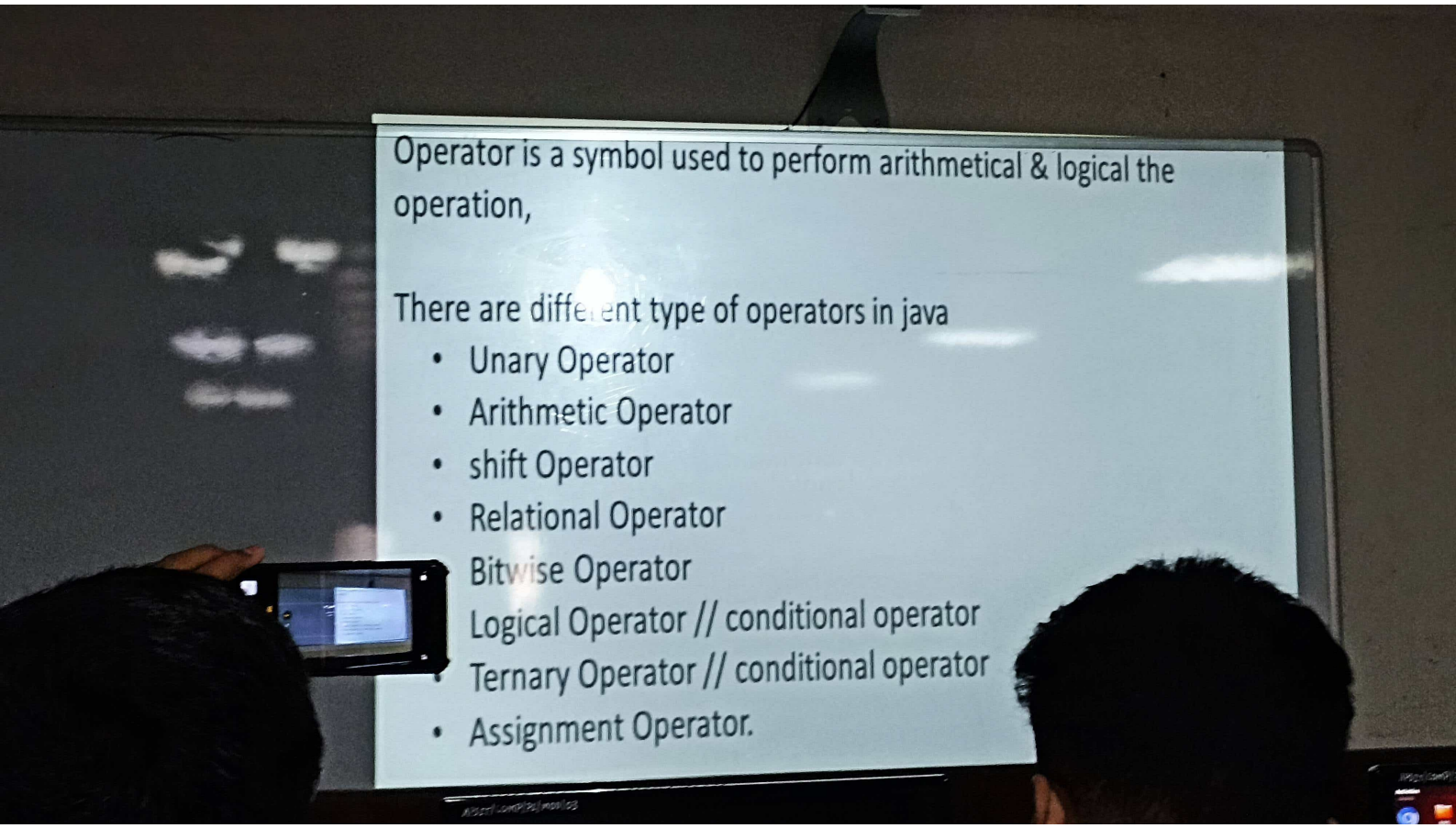
Java program compilation and execution process




1 i=20
2 i++ 20
3 ++i 22
4 i 22
5 ++i 23
6 ++i 24
7 i 2
8 i++ 24
9 ++i 26
AP25EI/COMP/PE/MON/03



	Category	Precedence
Unary Operator	postfix	expression++ expression--
	prefix	++expression --expression +expression -expression ~!
Arithmetic Operator	multiplication	* / %
	addition	+ -
Shift Operator	shift	<< >> >>>
Relational Operator	comparison	< > <= >= instanceof
	equality	== !=
Bitwise Operator	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
or	logical AND	&&
	logical OR	||
ernary Operator	ternary	? :
ssignment erator	assignment	= += -= *= /= %= &= ^= |= >>= >>>=


Operator is a symbol used to perform arithmetical & logical the operation,
There are different type of operators in java
Unary Operator
Arithmetic Operator
shift Operator
Relational Operator
Bitwise Operator
Logical Operator // conditional operator
Ternary Operator // conditional operator
Assignment Operator.

Why character (char) use 2 byte in java and what is \uoooo?

Ans: Java support 16+ international languages and 1 byte is not sufficient to st e all characters.

Important Note:

- To represent numeric values (10,20,30...etc) use **byte, short, int, long.**
- To represent decimal values(floating point values 10.5,20.5...etc) use **float, double.**
- To represent character use **char** and take the character within single quotes.
- To represent true, false use **boolean.**
- Except **boolean** and **char** remaining all data types consider as a signed data types because we can represent both +ve & -ve values.
- Float will give 5 to 6 decimal places of accuracy but double gives 14

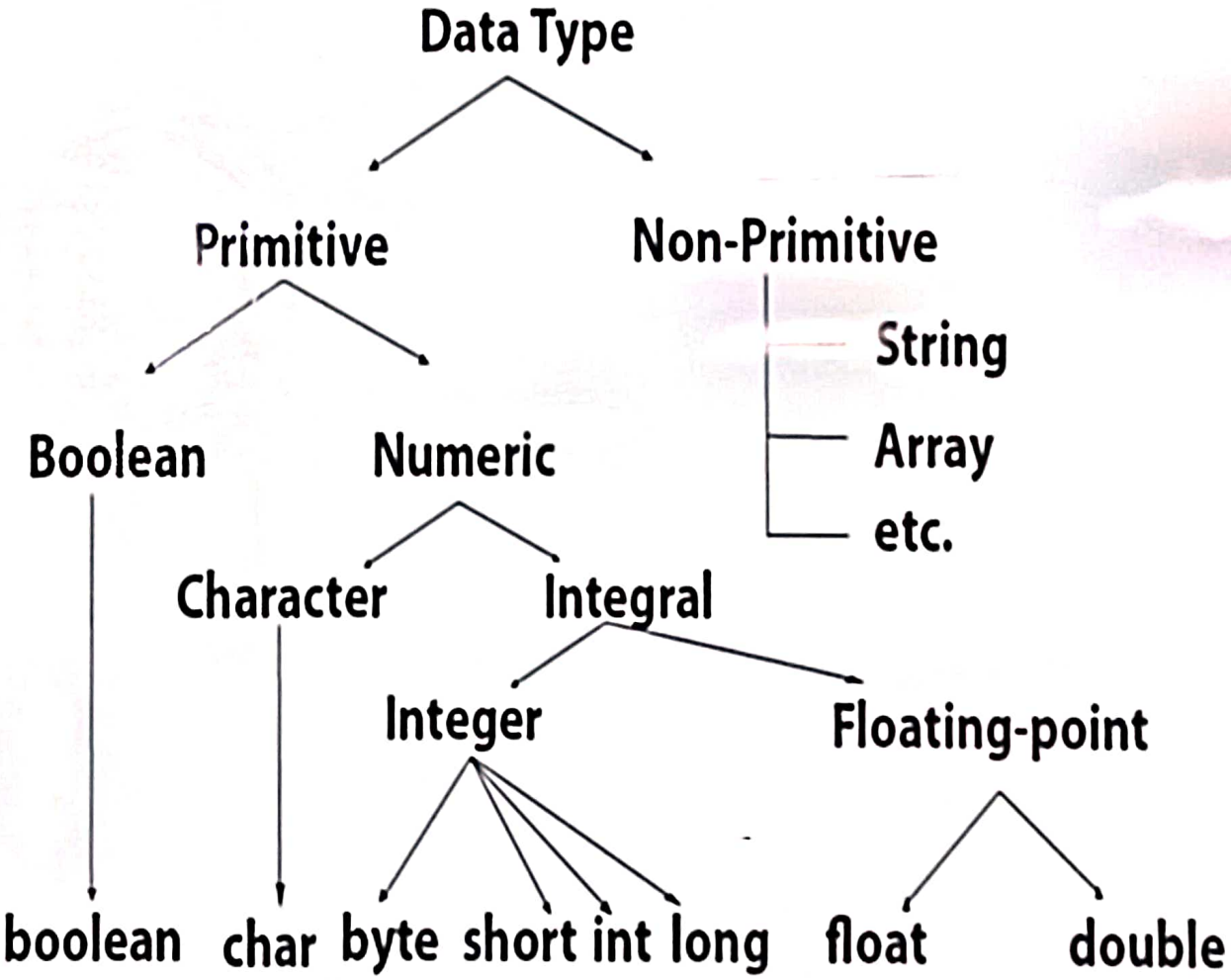
There are 8 primitive data types in java

Data Type	Size	Range	Default values
byte	1 Byte	-128 to 127	0
short	2 Byte	-32768 to 32767	0
int	4 Byte	-2147483648 to 2147483647	0
long	8 Byte	-9,223,372,036,854,775,808 to 9 ,223,372,036,854,775,807	0
float	4 Byte	-3.4e38 to 3.4e	0.0
double	8 Byte	-1.7e308 to 1.7e308	0.0
char	2 Byte	0 to 65535	single space
Boolean	no-size	no-range	false upto

int – upto 9 digit long – upto 16 digit

Default datatype in Integer Group: int

Default datatype in Floating Group: double


Data Type
Primitive
Non-Primitive
String
Array
etc.
Boolean
Numeric
Character
Integral
Integer
Floating-point
boolean
char
byte
short
int
long
float
double

Data type tells what type of value hold by the variable.

- Each and every variable and expression has type
- Strongly typed programming language
- Data types are representing how much memory is allocated for variable

There are two types of data types in java

1. Primitive data types
2. Non-primitive data types

Constant variables:

Constant variables should be in **all uppercase** with words separated by underscores ("_").

Examples:

```
static final int WIDTH = 4;
```

Packages:

Package name is always written in **all-lowercase ASCII letters** and should be one of the top-level domain names, like com, edu, gov, mil, net, org.

Examples:

```
com.sun.eng                com.help4code.programs
java.lang                  java.util
```

Methods:

Methods should be **verbs** and the first letter lowercase and with the first letter of each internal word capitalized.

Examples:

```
void change();          void changeGear(int newValue);
void applyOnBrakes(int decrement);
```

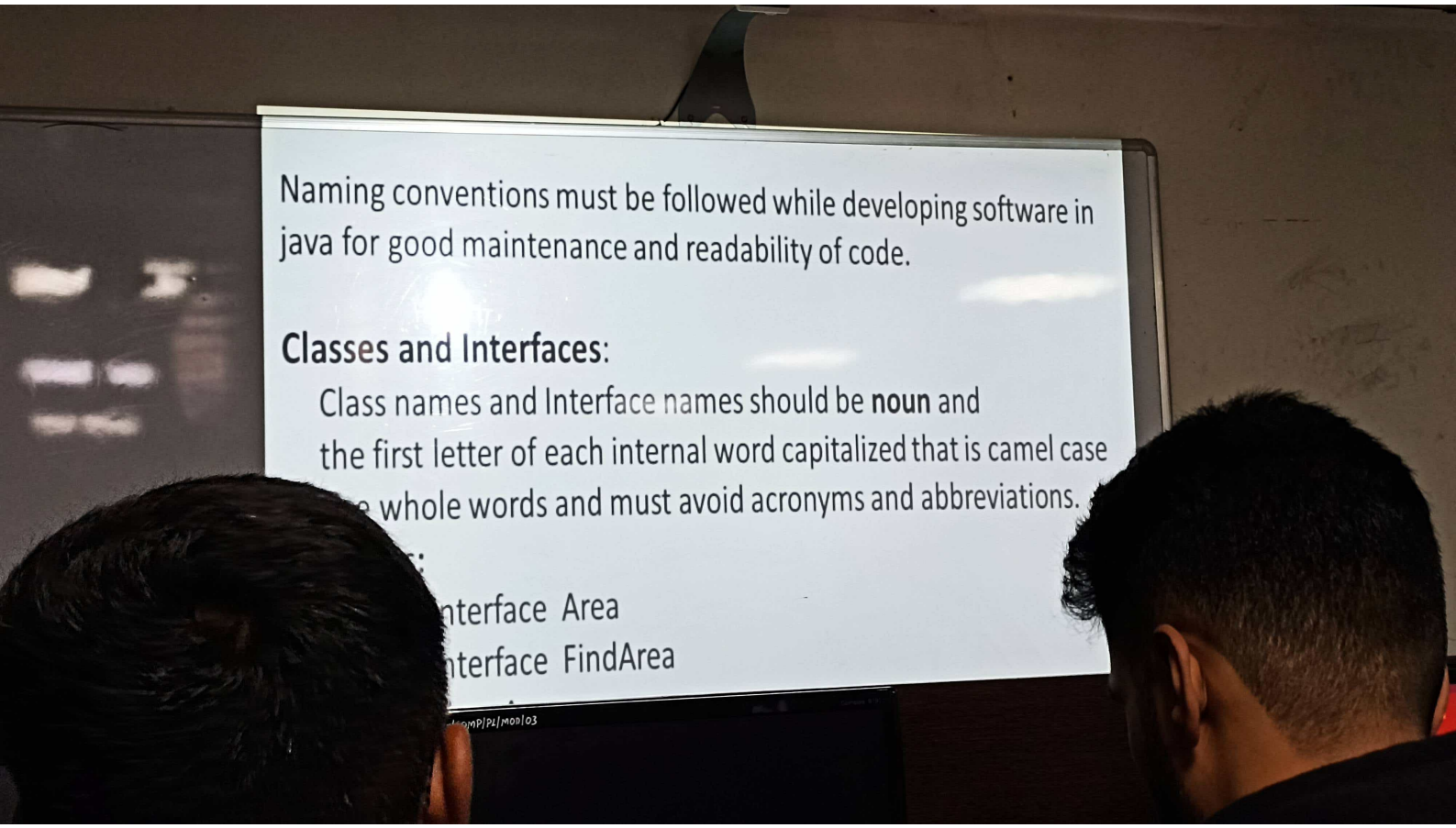
Variables and Object Names:

One-character variable names should be avoided except for temporary variables.

Variable names should be short yet meaningful and in small letters.

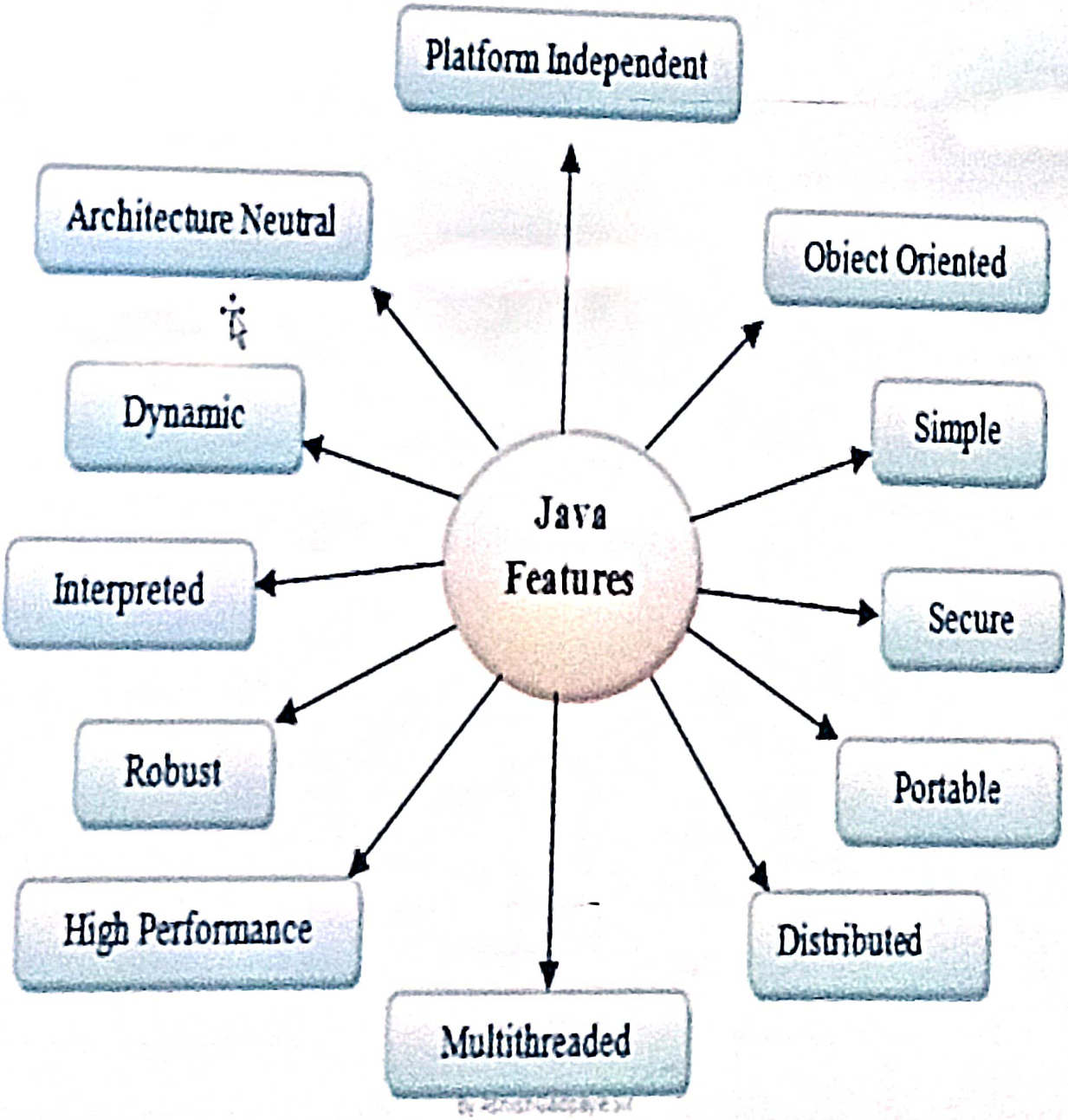
Examples:

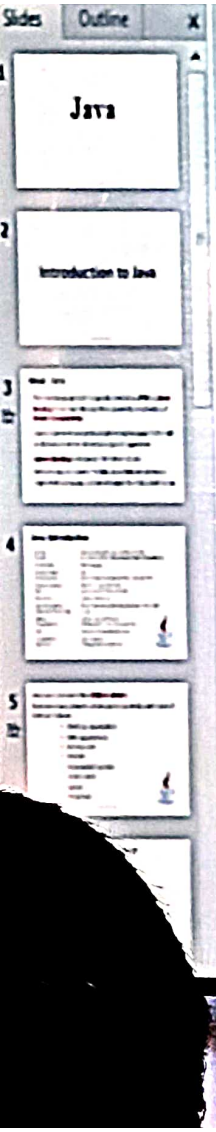
```
int speed = 0;          double areaOfCircle=55.3;
```


Naming conventions must be followed while developing software in java for good maintenance and readability of code.
Classes and Interfaces:
Class names and Interface names should be noun and
the first letter of each internal word capitalized that is camel case
whole words and must avoid acronyms and abbreviations.
nterface Area
nterface FindArea

Java Features

Class
Object
Abstraction
Encapsulation
Inheritance
Polymorphism
Exception Handling
Message Passing
Dynamic Binding

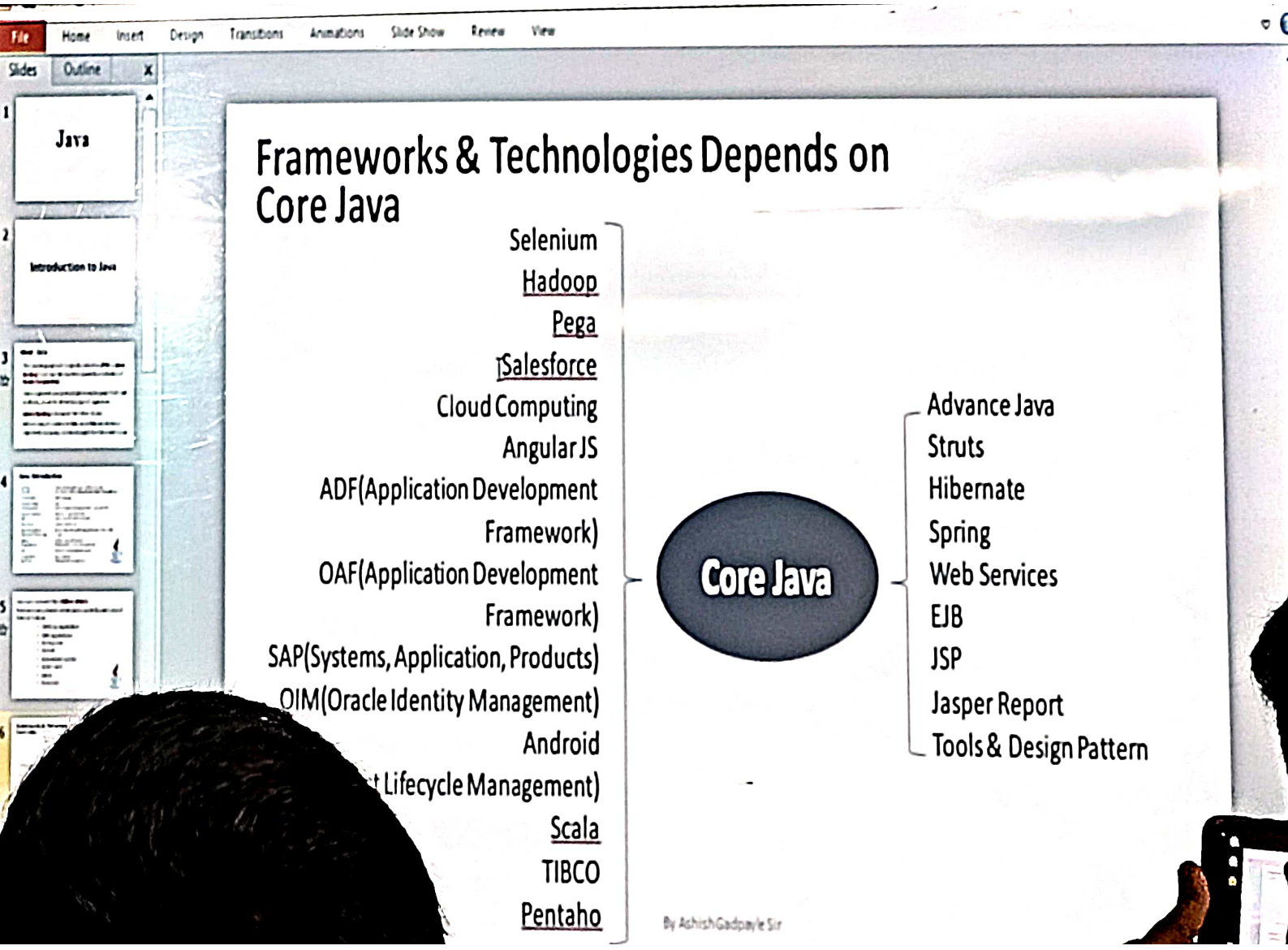

Java
Features
Platform Independent
Object Oriented
Simple
Secure
Portable
Distributed
Multithreaded
High Performance
Robust
Interpreted
Dynamic
Architecture Neutral



JAVA released to the market in three categories

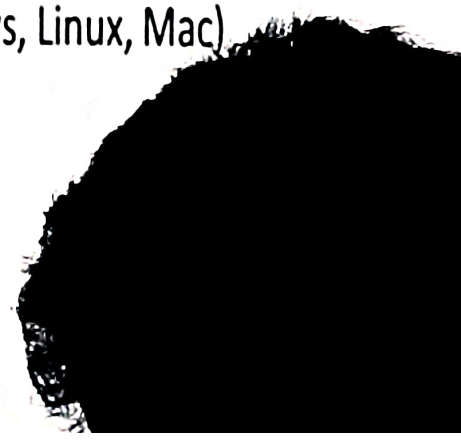
- J2SE (JAVA 2 Standard Edition)
- J2EE (JAVA 2 Enterprise Edition)
- J2ME (JAVA 2 Micro/Mobile Edition)

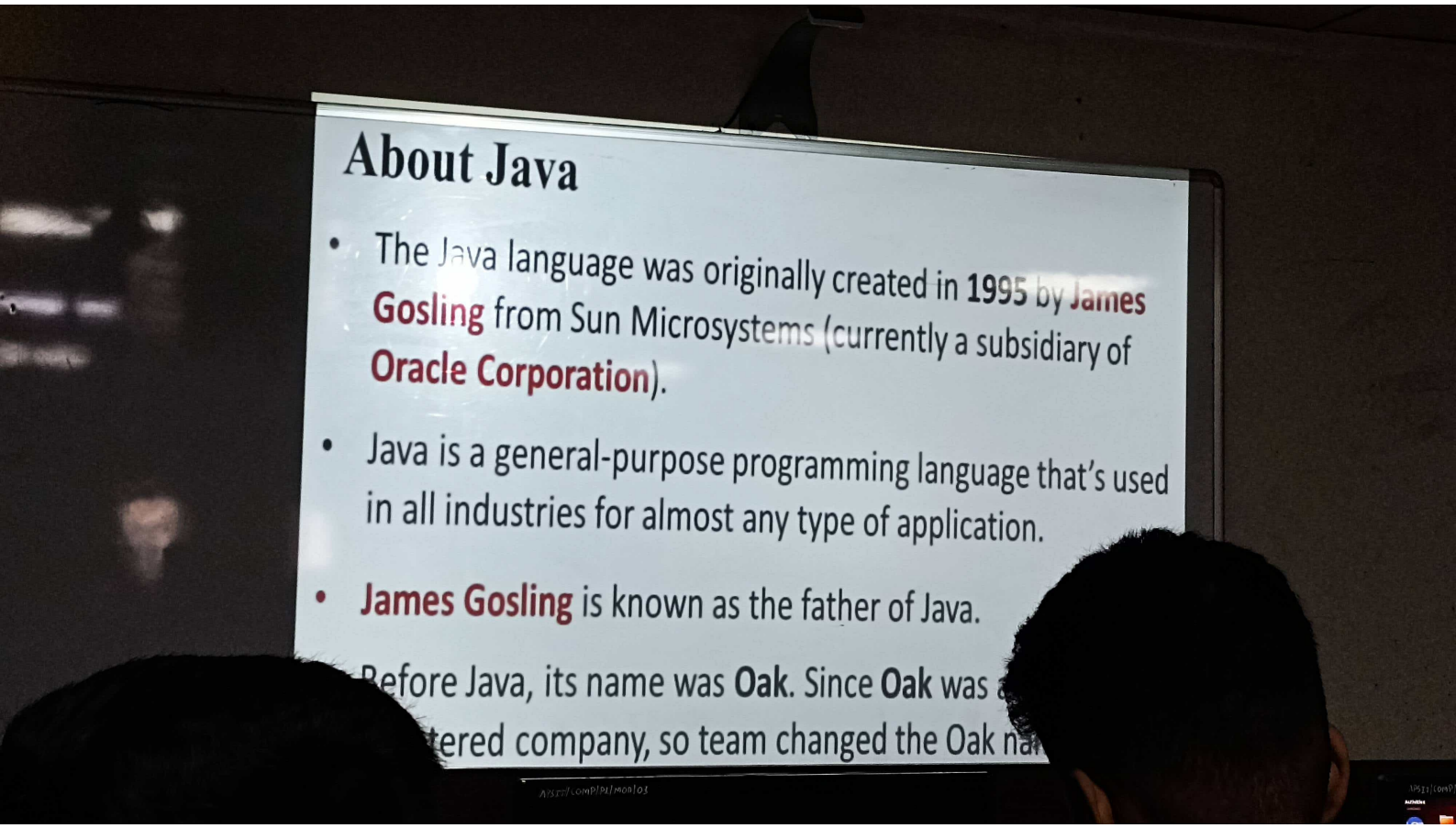
J2SE is used for developing client side applications/programs.
J2EE is used for developing server side applications/programs.
J2ME is used for developing embedded and mobile devices applications/programs.


File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Slides
Outline
Java
Introduction to Java
Frameworks & Technologies Depends on Core Java
Selenium
Hadoop
Pega
Salesforce
Cloud Computing
Angular JS
ADF(Application Development Framework)
OAF(Application Development Framework)
SAP(Systems, Application, Products)
OIM(Oracle Identity Management)
Android
Lifecycle Management)
Scala
TIBCO
Pentaho
Core Java
Advance Java
Struts
Hibernate
Spring
Web Services
EJB
JSP
Jasper Report
Tools & Design Pattern
By AshishGadpayle Sir

Java Introduction

Author	:	James Gosling (Known as Father of Java)
Vendor	:	Sun Micro System **(Now Under Oracle Corporation)**
Initial Name	:	OAK language
Present Name	:	Java
Initial version	:	jdk 1.0 (java development kit) - January 1996
Present version	:	JDK 21 – Sep, 19th 2023
Type	:	open source & free software
Extensions	:	.java & .class & .jar
Operating System	:	Multi Operating System(eg Windows, Linux, Mac)
Implementation Lang	:	c, cpp
Symbol	:	coffee cup with saucer
Slogan/Motto	:	WORA(write once run anywhere)
		Stanford Universally Network
		java compiler




About Java
• The Java language was originally created in 1995 by James Gosling from Sun Microsystems (currently a subsidiary of Oracle Corporation).
• Java is a general-purpose programming language that's used in all industries for almost any type of application.
• James Gosling is known as the father of Java.
Before Java, its name was Oak. Since Oak was
tered company, so team changed the Oak