

File Edit View

07_10_2022_DA_Discussion_notes - Notepad

what is difference between MLL and byte code?

MLL -> contains instructions in 0's and 1's

Bytecode -> It contains instructions neither in HLL nor in MLL

what is architectural neutral?

Write code once in any platform and run the code in any other platform without worrying its architecture.

Architectural neutral -> Write Once Run Anywhere(WORA)

what is jit?

JIT -> Just In Time Compiler(reference to JDK Architecture)

jvm comes between the platform to make independent...where c sticks...
on which level, briefly explain please

6

jvm comes between the platform to make independent...where c stuck...
on which level. briefly explain please

C language -----> compiler-----> .obj file(contains instructions in MLL)
JavaLanguage--> JavaCompiler -->.classfile(contains instructions neither in MLL nor in HLL)

.obj file -> windows(run)
-> linux(no output)
-> mac(no output)

Architectural Neutral/Platform independence/WORA

.classfile -> JVM(windows)
-> JVM(linux)
-> JVM(mac)

can u explain about the how the jvm works?

Un 28 Cpl 3 24°C
mostly cloudy



100% Windows (CELT) UTF-8

8

Eng

IN

07.10.2022

File Edit View
.classfile -> JVM(windows)
-> JVM(linux)
-> JVM(mac)

can u explain about the how the jvm works?

.class file(byte code) -> JVM (reads bytecode in interpretation style)

sir jvm is platform dependent?
yes jvm is platform dependent.

sir wht is obj oriented lang?

Obj oriented language means writing the code using the principles of OOPS.
eg: inheritance, polymorphism, abstraction,...

If we compiled our app in jdk18 and uploaded it in the cloud. Will it cause problems in the future after it is not supported?

Un 36 Cpl 1
23°C
mostly cloudy



File Edit View
07_10_2022_QA_Discussion_oops - Notepad

sir wht is obj oriented lang?

Obj oriented language means writing the code using the principles of Ooops.
eg: inheritance,polymorphism,abstraction,...

Q>

If we compiled our app in jdk18 and uploaded it in the cloud. Will it cause problems in the future after it is not supported?

No probs becoz java provides backward compatibility.

Q> jvm contains java compiler or java compiler is seperate?

JDK => Compiler + JRE

|=> JVM + library tools

Developers[JDK] -> write the code ,test the code and run the code.

Enduser[JRE] -> just run the code



Un 47 Cgi 17
23°C
Mostly cloudy



100% Windows (C:\UF) UTF-8

8

100%

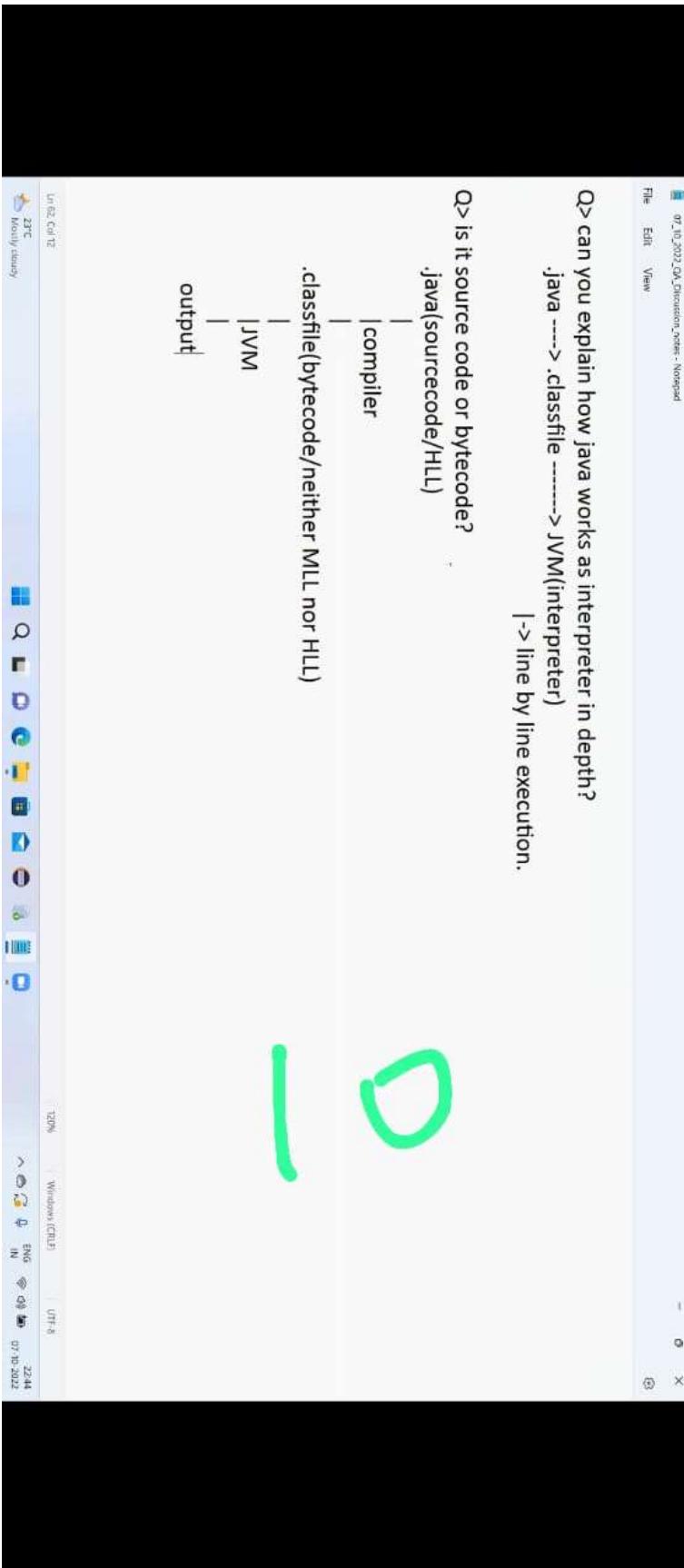
Windows (C:\UF)

UTF-8

ENG

IN

07_10_2022



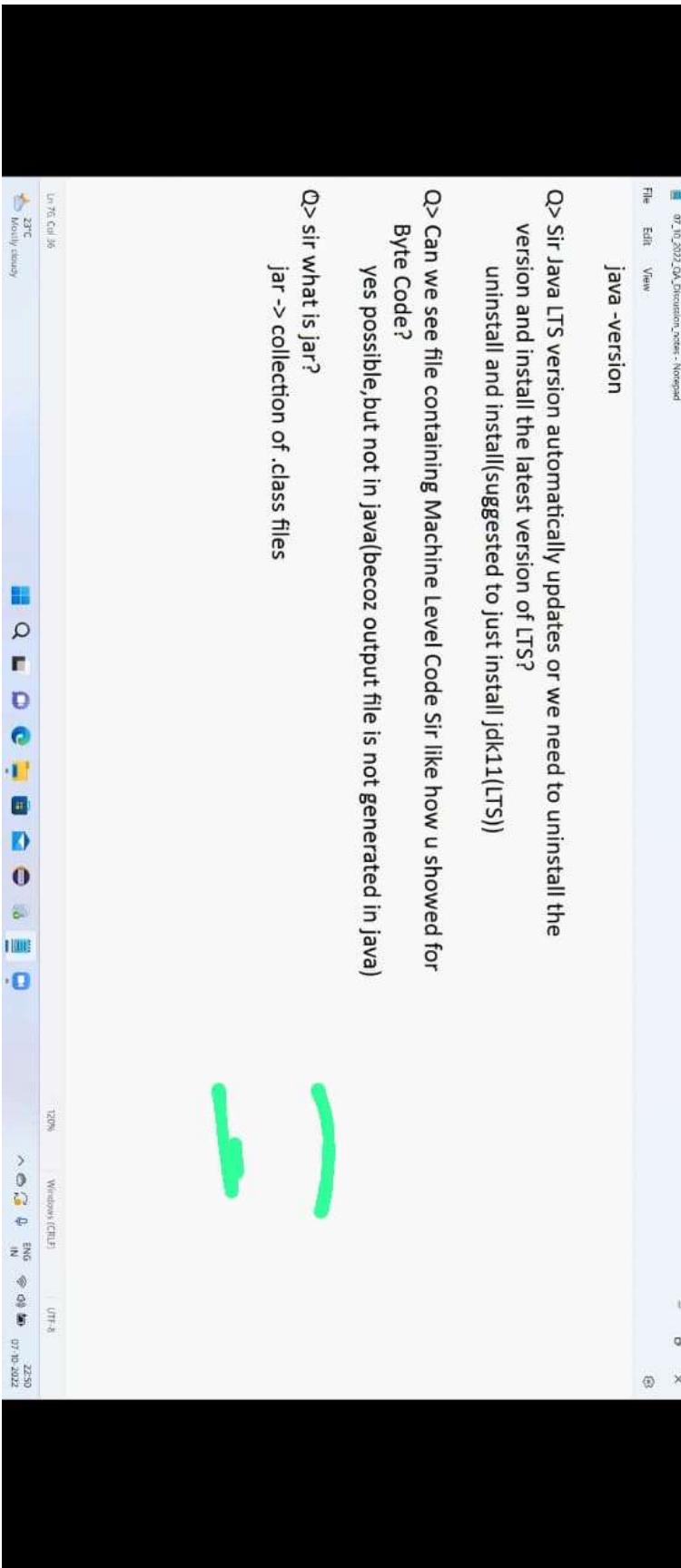
Q> can you explain how java works as interpreter in depth?

.java ----> .classfile -----> JVM(interpreter)

| -> line by line execution.

Q> is it source code or bytecode?

```
java(sourcecode/HLL)
|
| compiler
|
|.classfile(bytecode/neither MLL nor HLL)
|
| JVM
|
| output
```



07_10_2022_QA_Discussion_notes - Notepad

File Edit View

```
java -version
```

Q> Sir Java LTS version automatically updates or we need to uninstall the version and install the latest version of LTS?

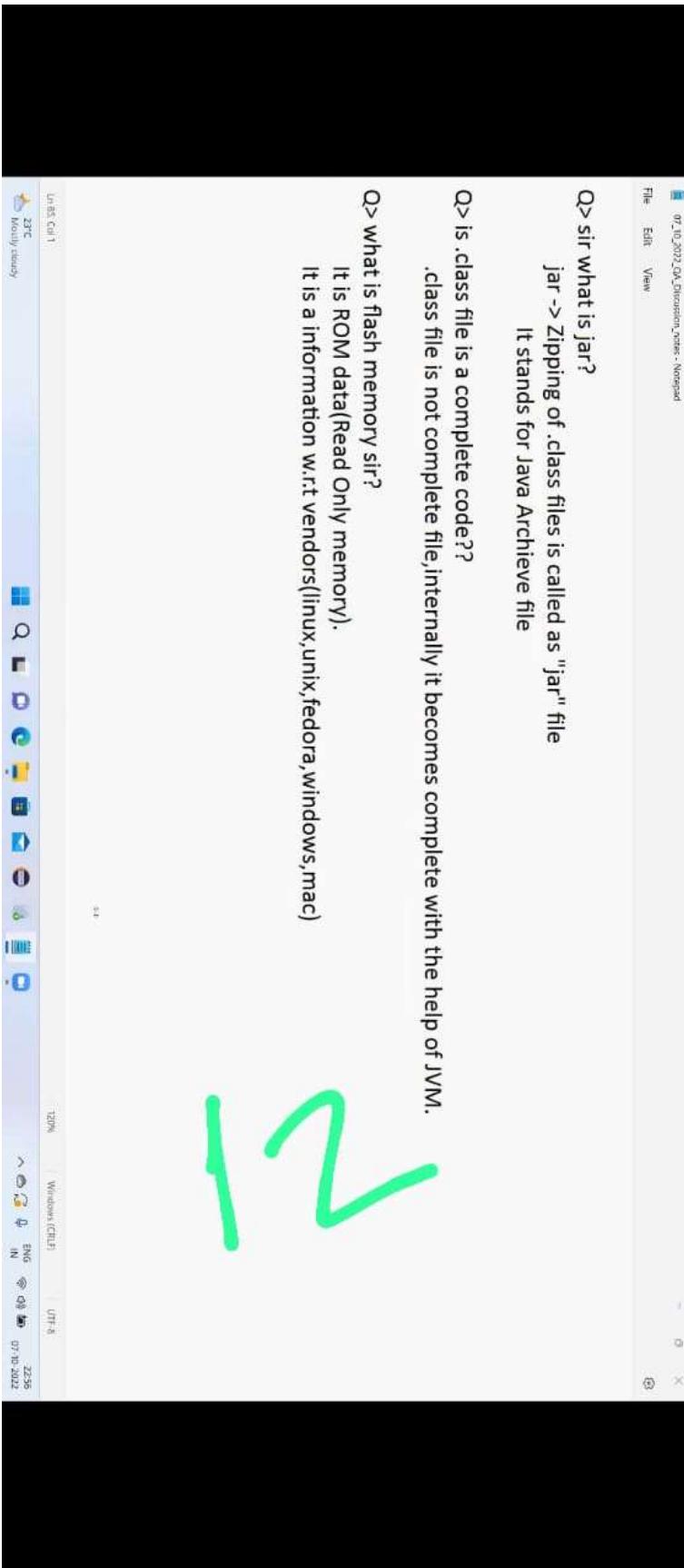
uninstall and install(suggested to just install jdk11(LTS))

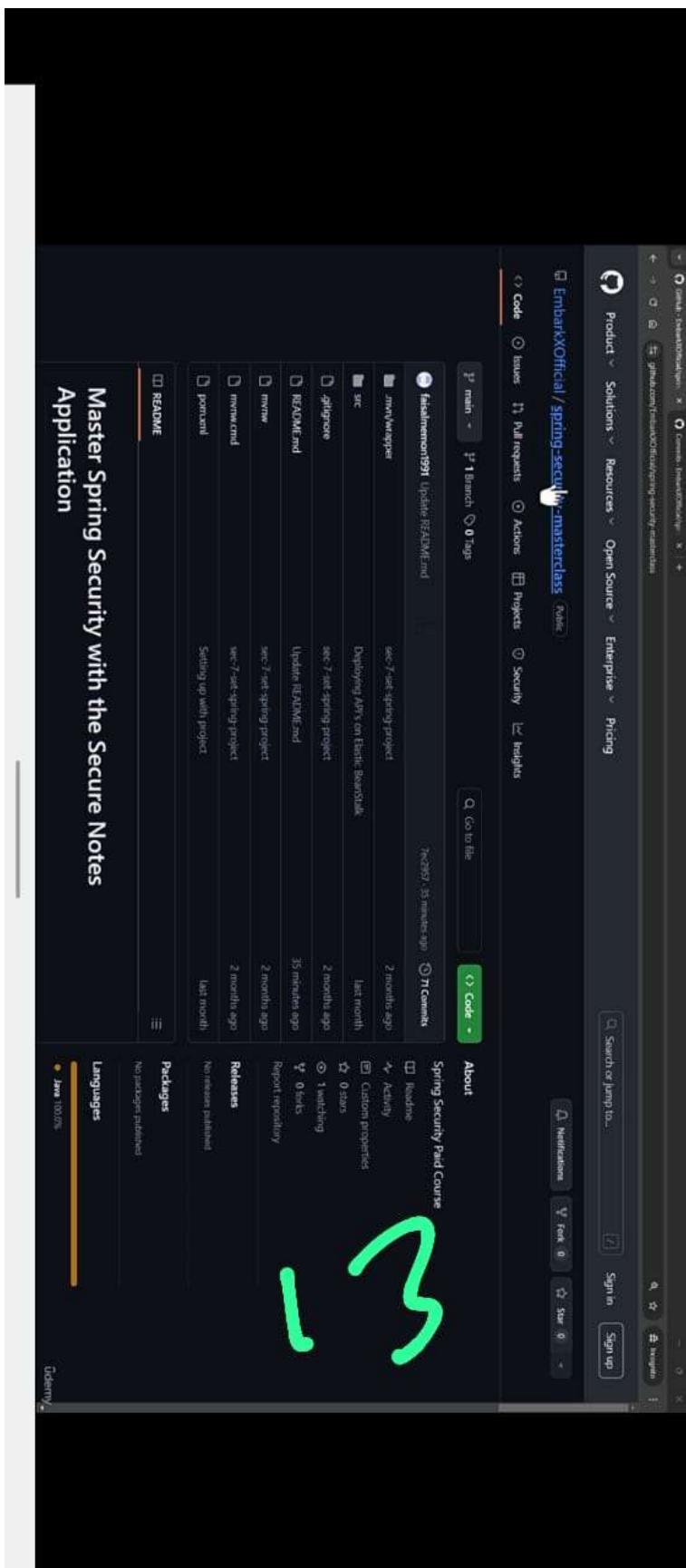
Q> Can we see file containing Machine Level Code Sir like how u showed for Byte Code?

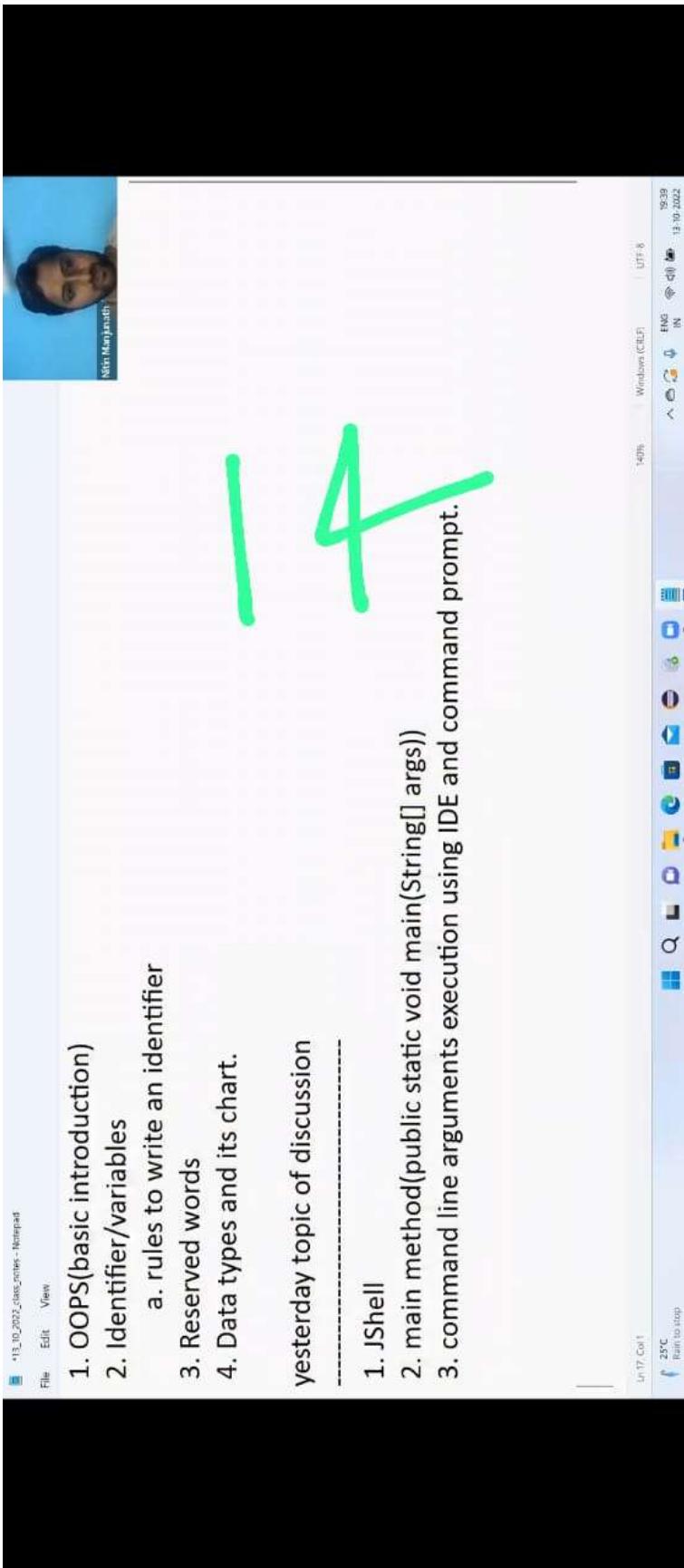
yes possible, but not in java(becoz output file is not generated in java)

Q> sir what is jar?

jar -> collection of .class files







OOPS

It stands for Object Orientation Principles.

Object -> real time instance.

eg: Car,Student,Employee

Every object in realtime will have 2 parts

what it has
what it does

eg: Car

name
noOfWheels
model
speed

13_10_2022 Classmate - Notepad

File Edit View

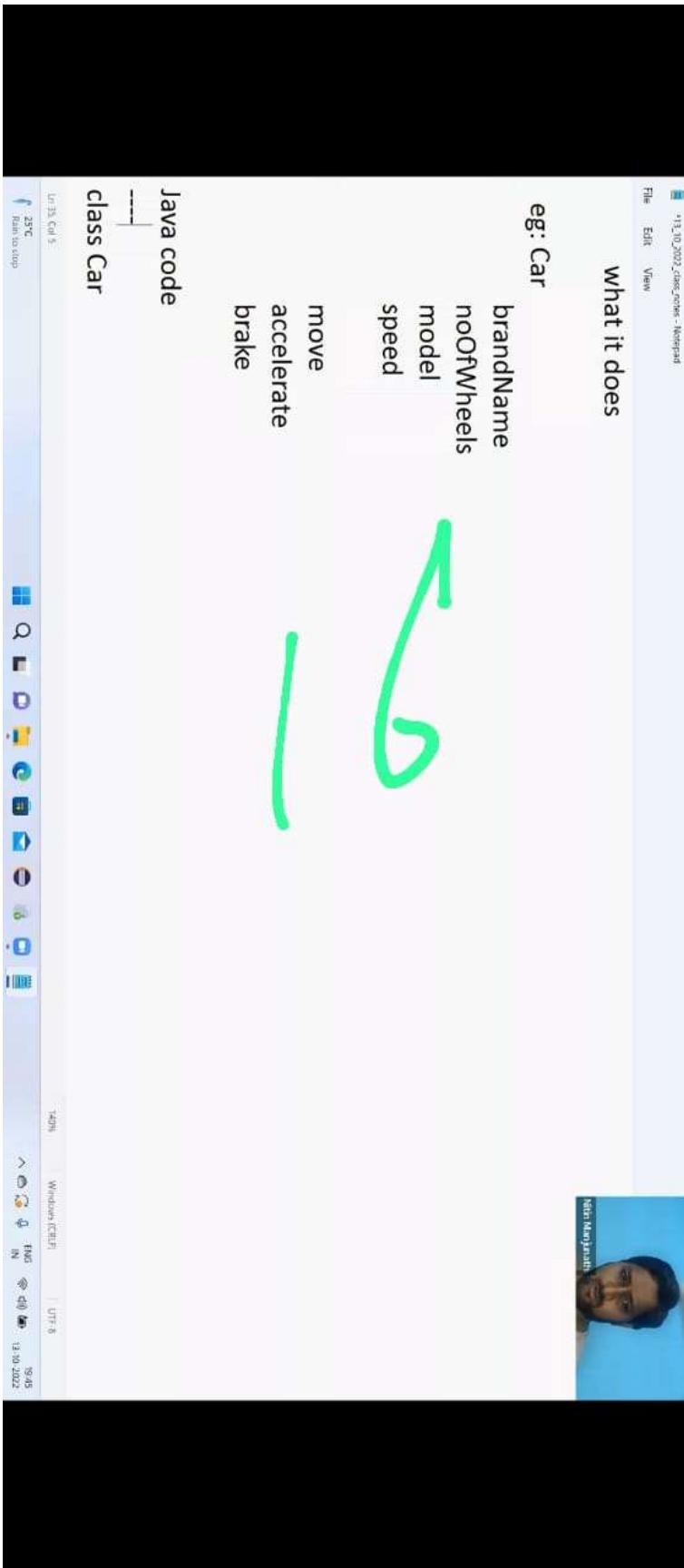
25°C Rain to stop

Windows (C:\UF)

UTF-8

140%

^ C IN ENG ⌂ 13-10-2022



```
13.10.2022 class-classes - Interfaz  
File Edit View  
class Car  
{  
    //HAS part of an Object is represented as a "variable".  
    String brandName;  
    int noOfWheels;  
  
    //DOES part of an object is represented through "methods"  
    public void move()  
    {  
        //logic of moving a vehicle  
    }  
  
    public void accelerate()  
    {  
        //logic of accelerating a vehicle  
    }  
}
```


14% Windows (CEP) UTF-8
25°C Rain to sun
União Civil 10
In 13.10.2022 19:54

Identifier

=====

It is a name in java program.

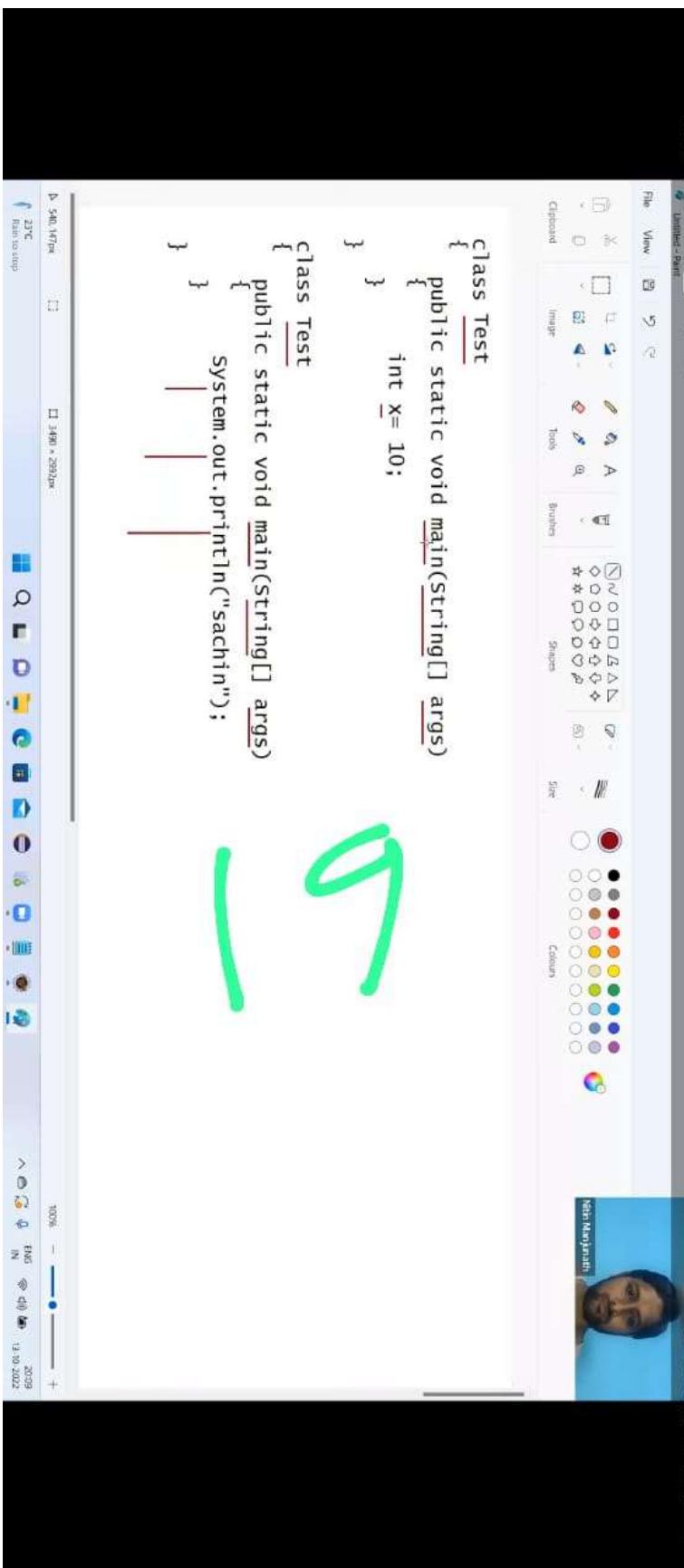
It can be a classname,methodname,variable name.

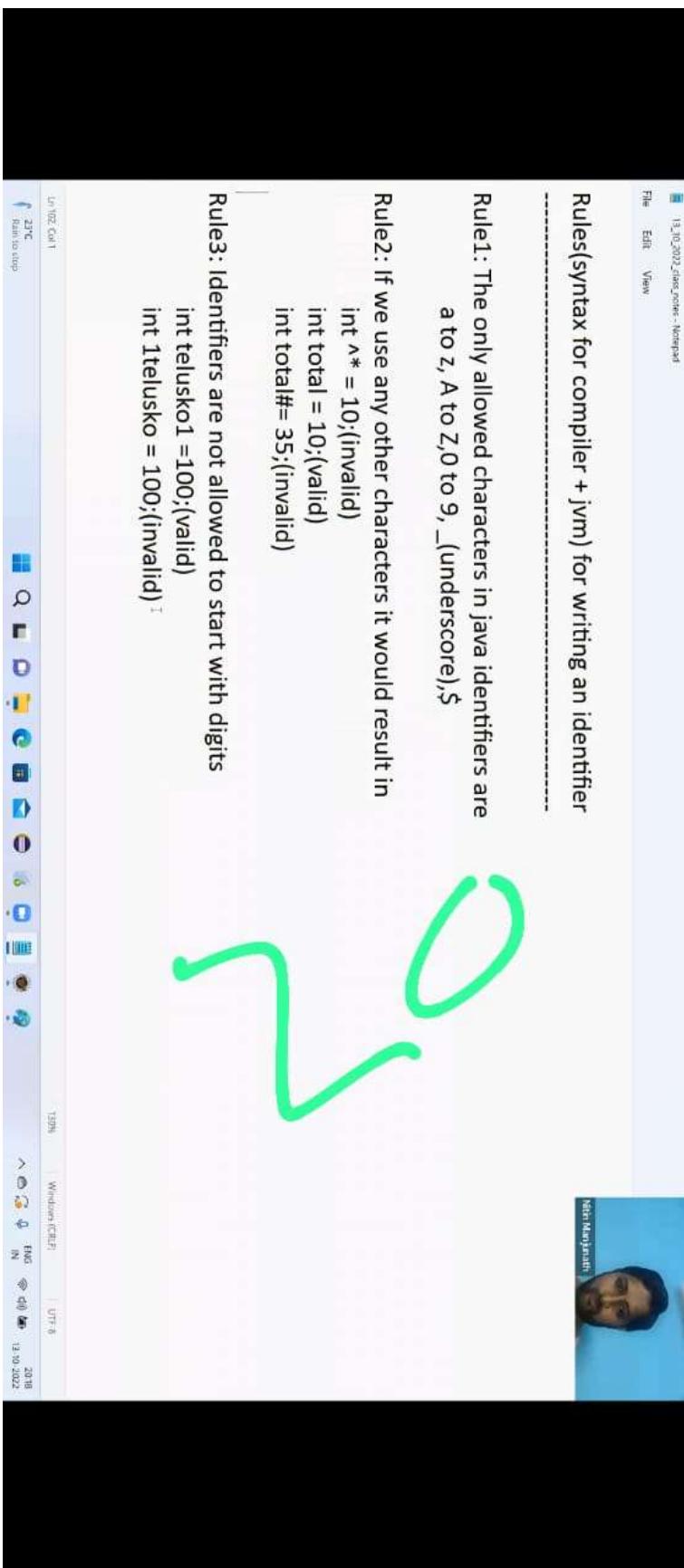
```
class Test{  
    public static void main(String[] args){  
        int x= 10;  
    }  
}
```



Lr 75 Col 42
22°C Rain to stop







13_2020/2020-05-20 - 10:00:00

Rule4: Java identifiers are case sensitive, meaning number and Number is different.

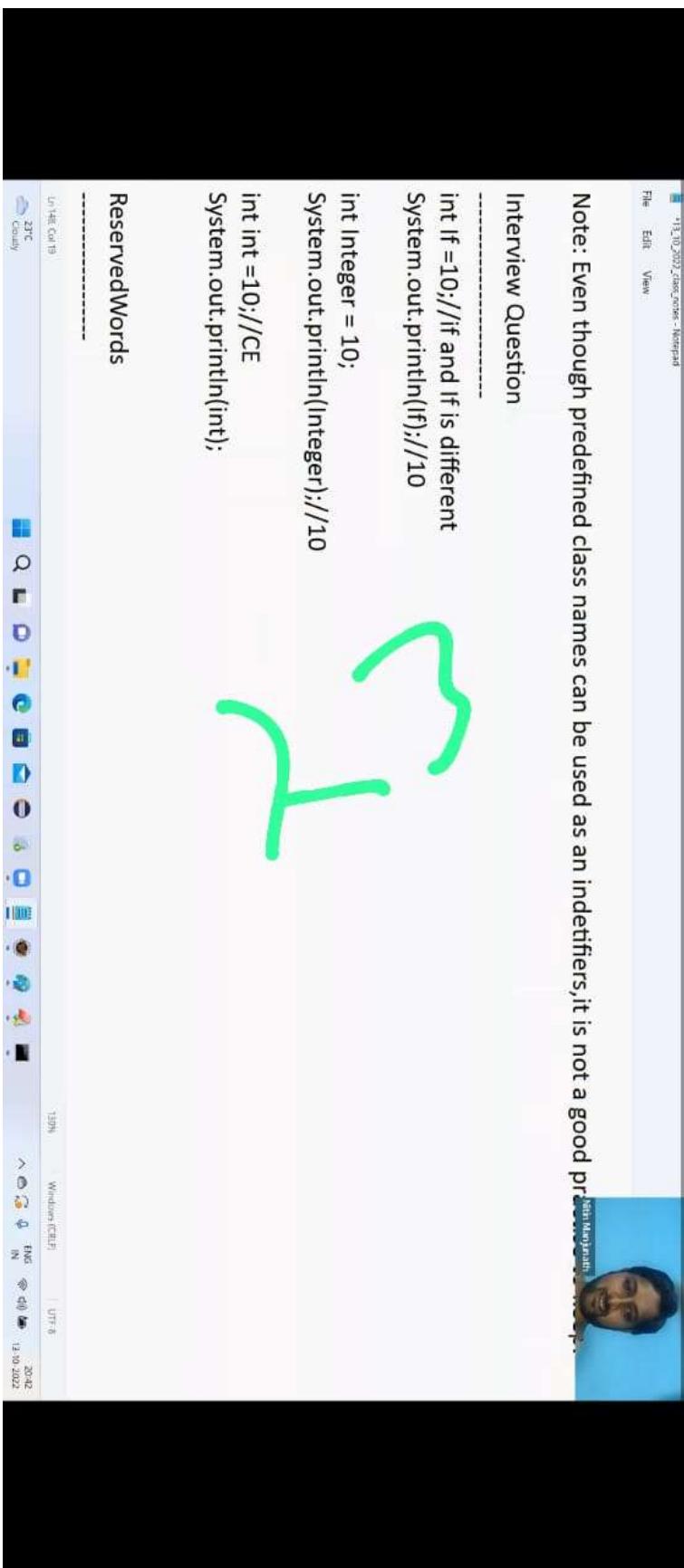
1

```
int number=10;  
int Number=20;  
int nUmber= 30;  
int NUMBER = 40;
```

Rule5: There is no length limit on java identifiers, but still it is a good practise to keep the length of the identifier not more than 15characters.

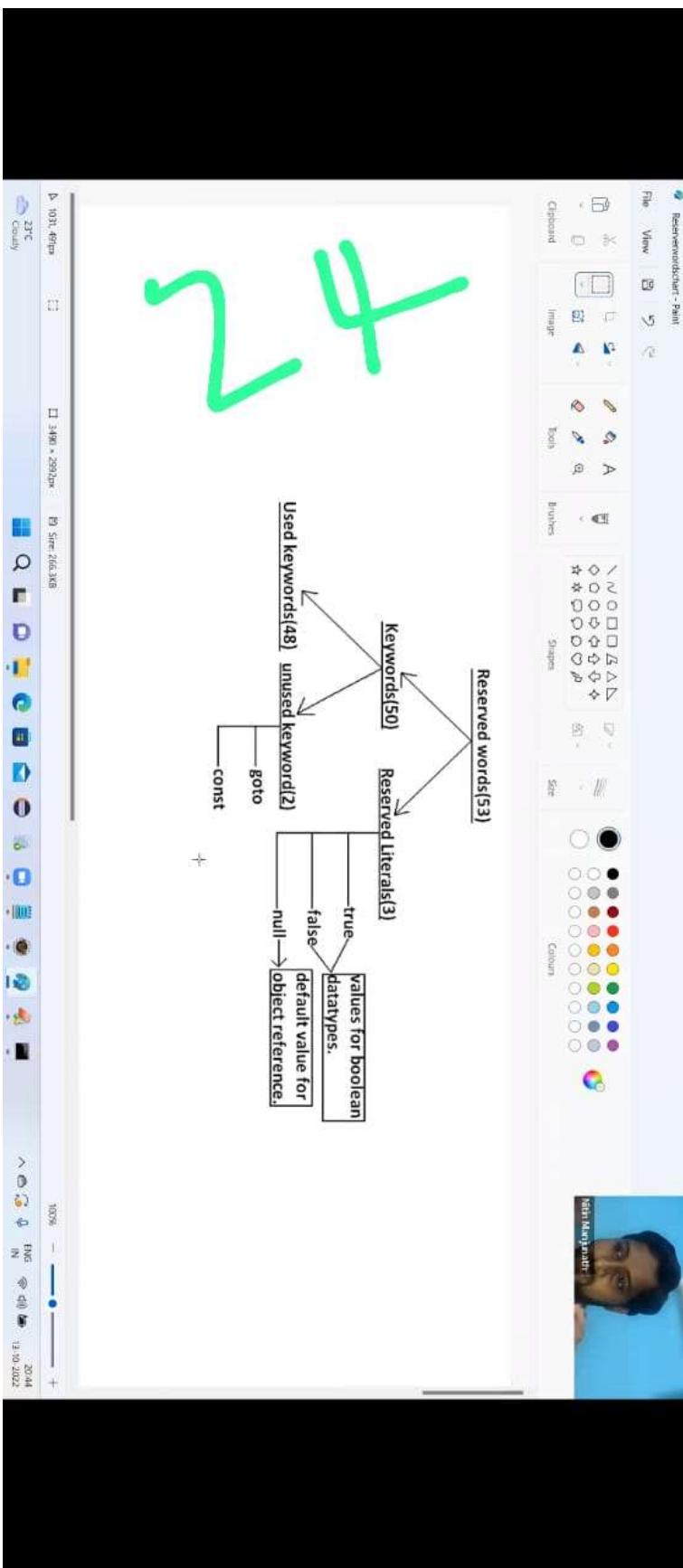


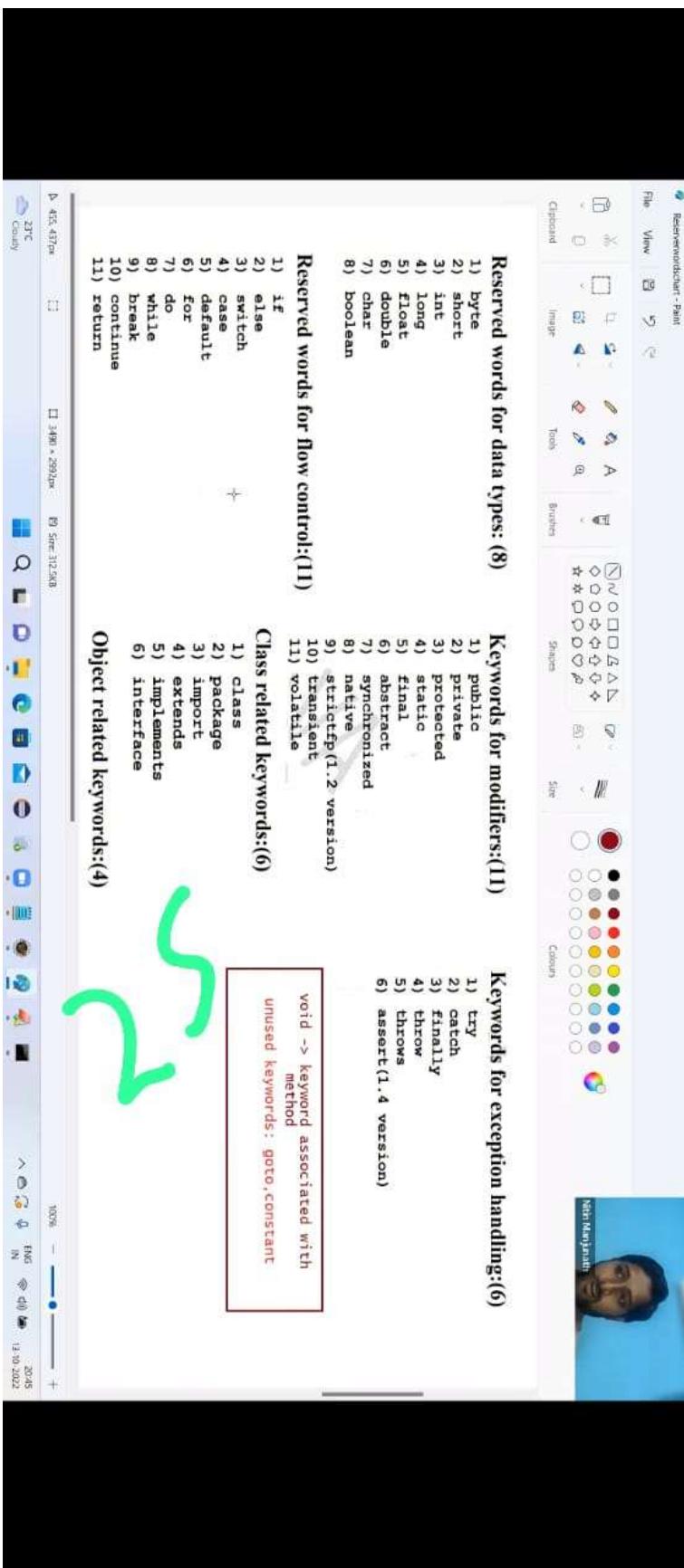
```
*13.10.2022.class_names - Notepad
File Edit View
Nitin Mangalwani
13.10.2022 23:54 112x601 130 KB
+13.10.2022.class_names - Notepad
File Edit View
Nitin Mangalwani
13.10.2022 23:54 112x601 130 KB
Rule6: We can't use reserve words as a identifiers.
eg: int if = 10; //CE
Rule7: Predefined class names can be used as identifiers.
eg#1
String Runnable = "sachin";
System.out.println(Runnable); //sachin
eg#2
int String = 10;
System.out.println(String); //10
Note: Even though predefined class names can be used as an indetifiers,it is not a good practise to keep
```

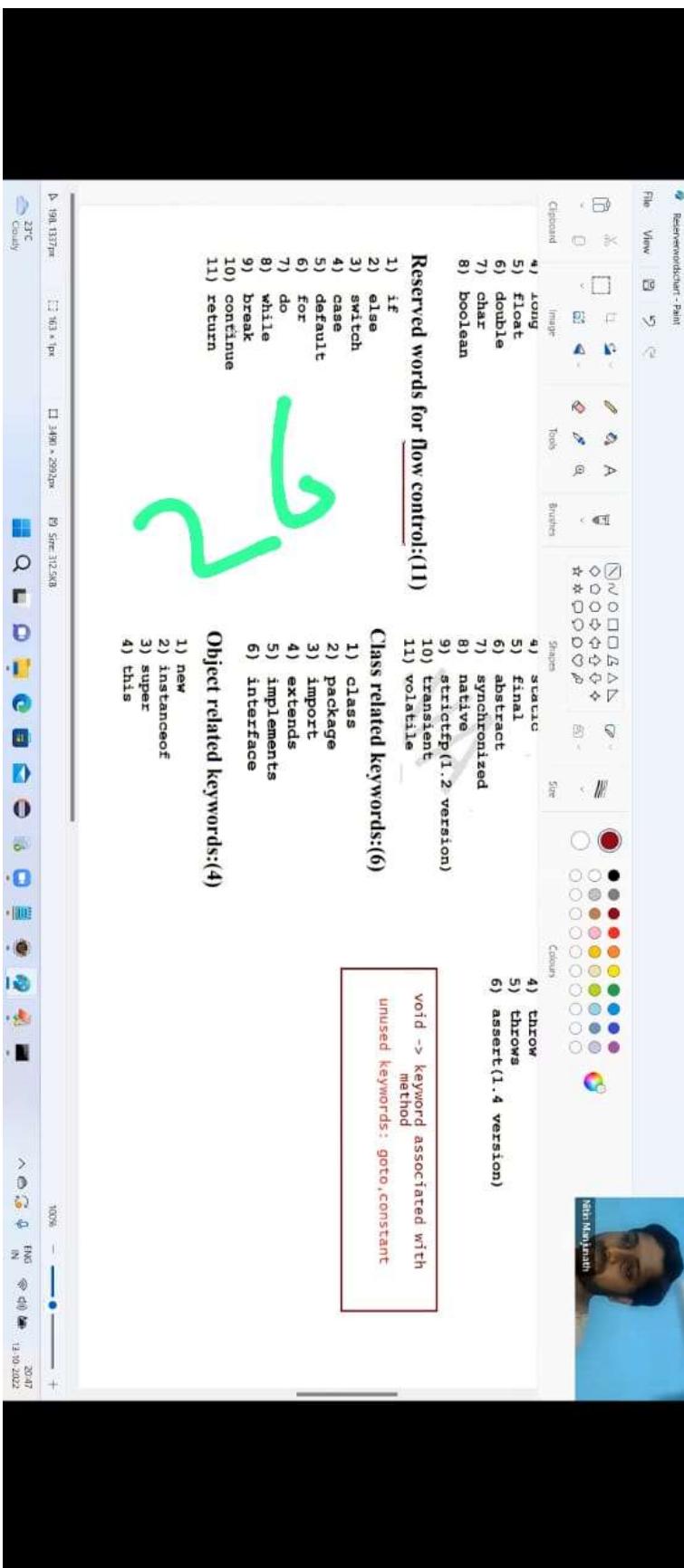


Un 146 Col 19
23TC
Cloudy

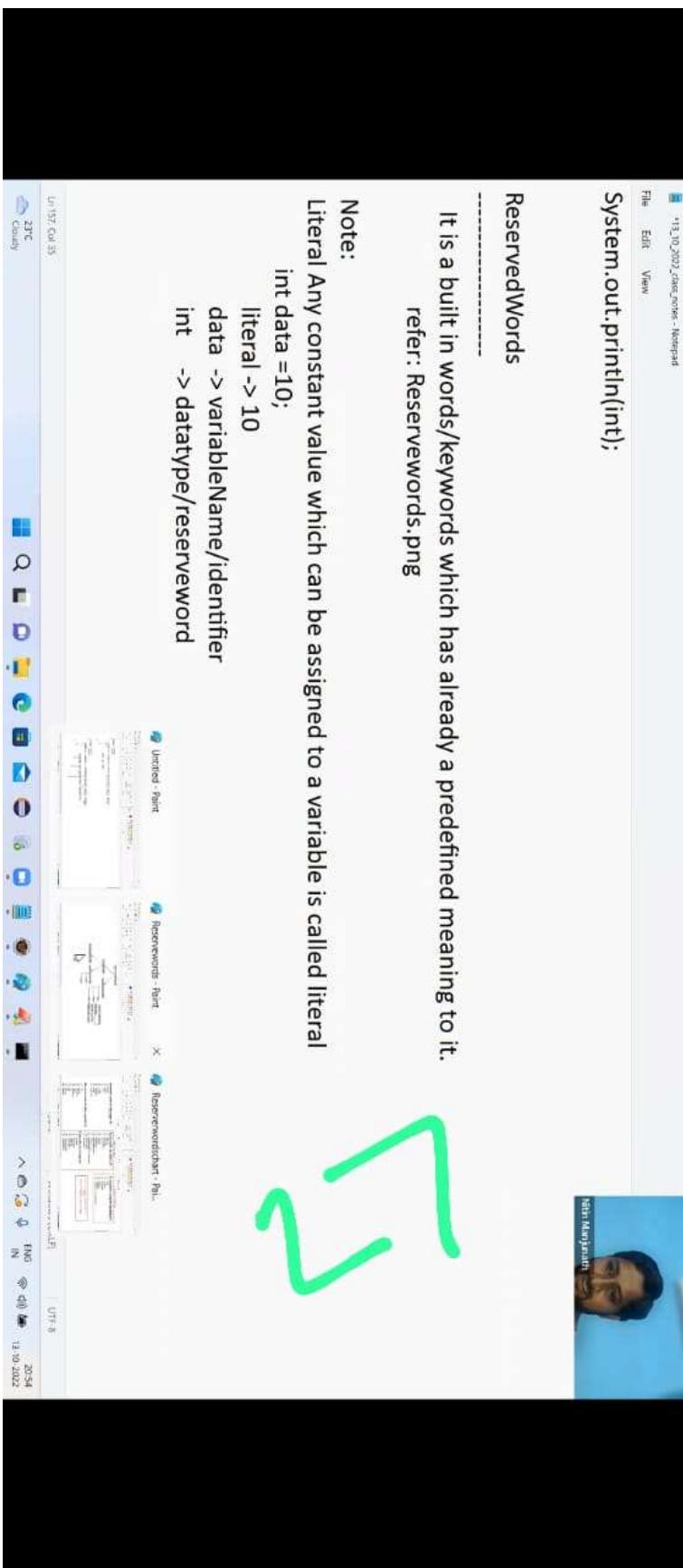
150% Windows (CE) | UTF-8 ENG (EN) 2042 13.10.2022



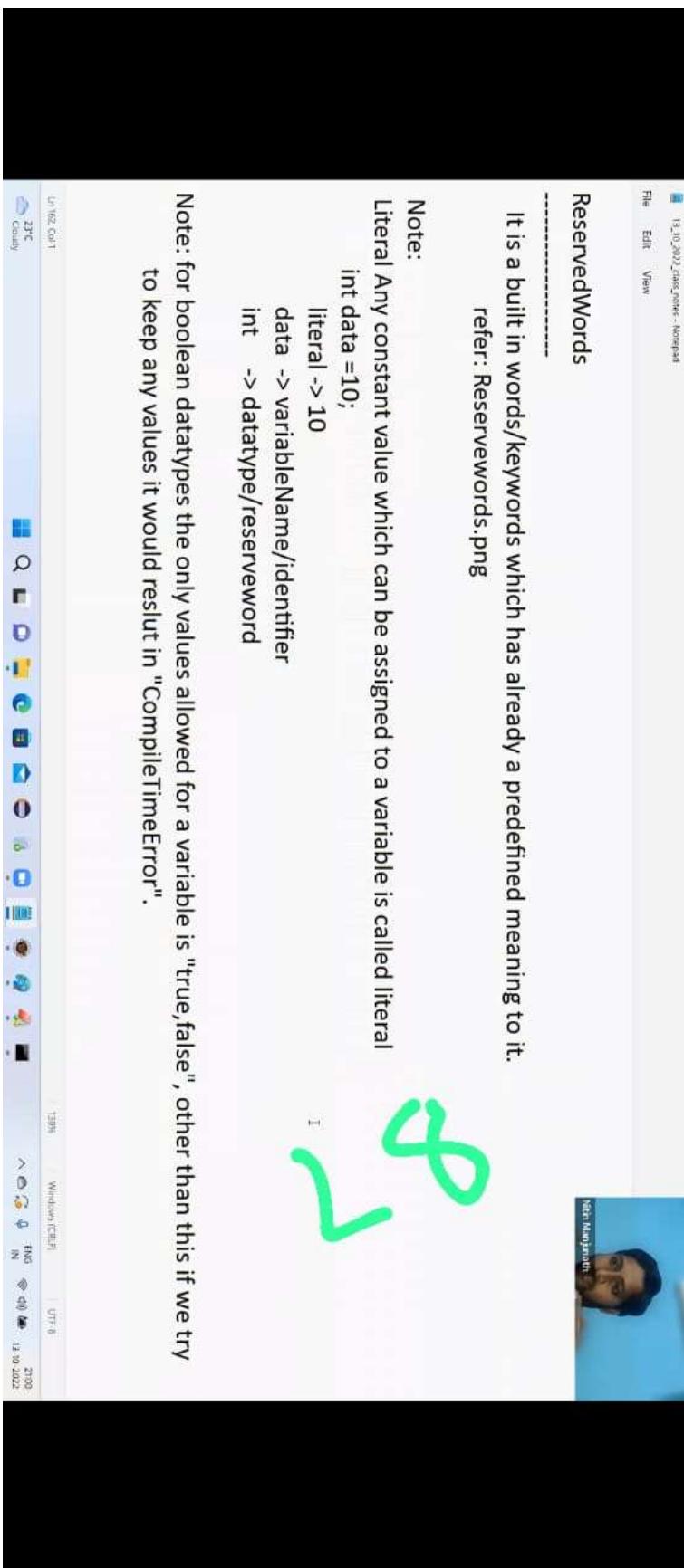




```
*13.10.2022.class_names - Notepad  
File Edit View  
System.out.println(int);  
  
ReservedWords  
-----  
It is a built in words/keywords which has already a predefined meaning to it.  
refer: Reservedwords.png  
  
Note:  
Literal Any constant value which can be assigned to a variable is called literal  
int data =10;  
literal -> 10  
data -> variableName/identifier  
int -> datatype/reserveword
```



23°C Cloudy



Note:
Literal Any constant value which can be assigned to a variable is called literal

```
int data =10;
literal -> 10
data -> variableName/identifier
int -> datatype/reserveword
```

Note: for boolean datatypes the only values allowed for a variable is "true/false", other than this if we try to keep any values it would result in "CompileTimeError".

13_10_2022_Gloss_Phrases - Notepad

File Edit View

literal -> 10
data -> variableName/identifier
int -> datatype/reserveword

Note: for boolean datatypes the only values allowed for a variable is "true/false", other than this if we try to keep any values it would result in "CompileTimeError".

Which of the following list contain only reserve words ?

1. final,finally,finalize

ans. finalize is not a reserveword, it is a method in Object class.

2. break,continue,return,exit

ans. exit is not a reserve word, it is a method in System class

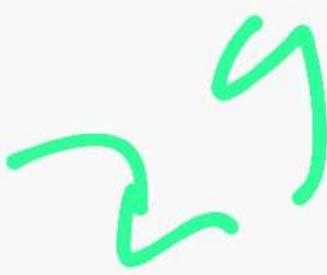
3. byte,short,integer,long

ans. Integer is not a reserve word,it is a predefined class

4. throw,throws,thrown

ans.

Un 173. Col B 24C Country 10% Windows (CEP) UTF-8 ENG (d) 13_10_2022 21:17



ans. thrown is not a reserve word,it is a userdefined variable.

Dataatypes

Every varaiable has a type,every expression has a type and all types are strictly typed/define in java becoz java is strictlytype /statically typed language.

Compiler role -> Compiler will check the value stored can be handled by datatype or not

This checking which is done by compiler is called "TypeChecking/Strictlytype checking".

Primitive datatypes

a.Numeric values
=> to store number

- a. whole number
- b. realnumber

b.character values
=> to store character type of data

c.boolean values



11:10, 10/2022, class_name - Notepad
File Edit View
Nitin Manjrekar

100% Windows (CEP) UTF-8
24C Cloudy 13:10 10/2022

13_10_2022_glossary - Notepad

File Edit View

Compiler role -> Compiler will check the value stored can be handled by datatype or not

This checking which is done by compiler is called "TypeChecking/Strictlytype check"

Primitive datatypes

=====

meaning -> data which is commonly used and supported by any language to store directly.

a. Numeric values
=> to store number

- a. whole number
- b. realnumber

b.character values
=> to store character type of data

c.boolean values
=> to store logical values



100% Windows (CEP) | UTF-8

24C Cloudy

13_10_2022 21:32

File Edit View

c.boolean values

=> to store logical values

Number data

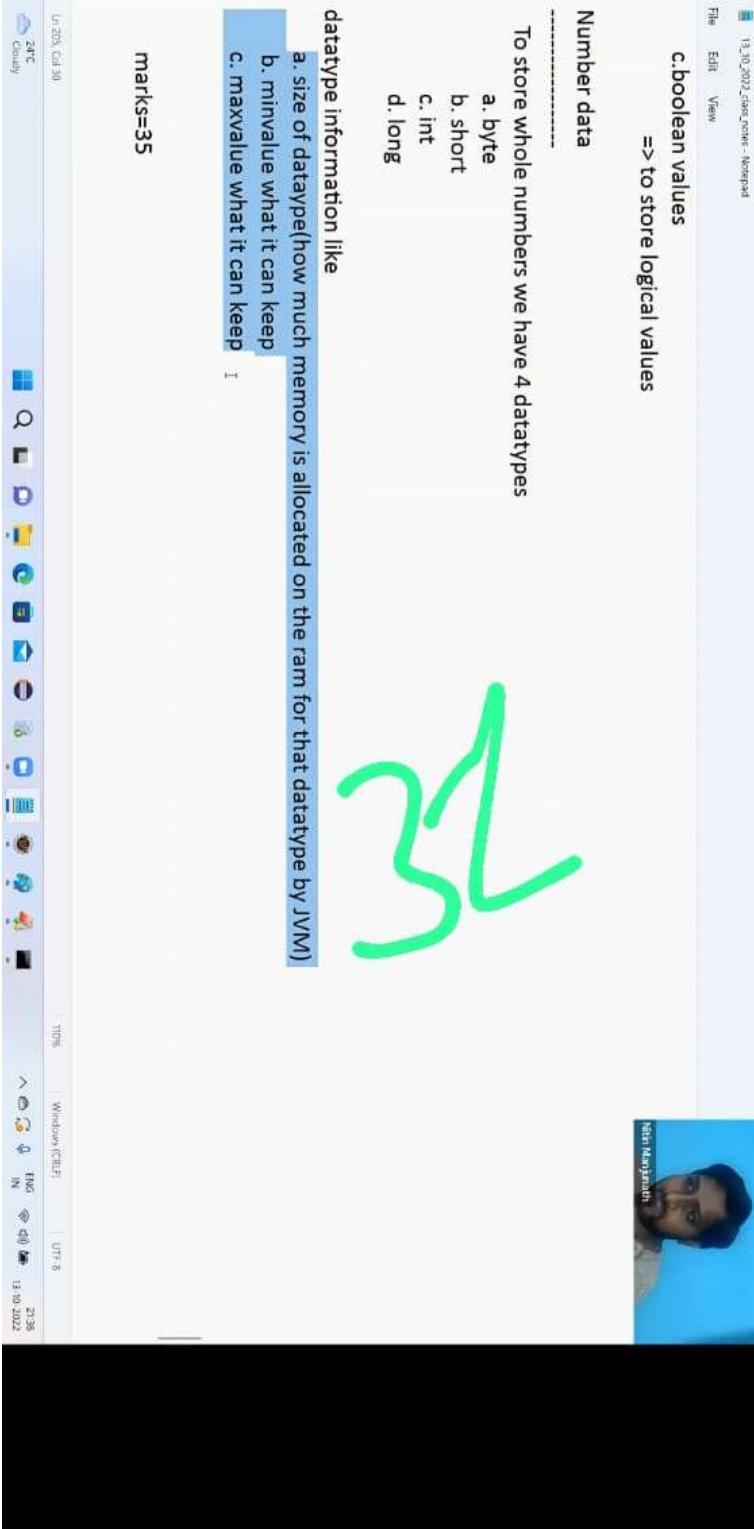
To store whole numbers we have 4 datatypes

- a. byte
 - b. short
 - c. int
 - d. long

datatype information like

- a. size or datatype(how much)
 - b. minvalue what it can keep
 - c. maxvalue what it can keep

marks=35



Command Prompt

```
symbol: variable MIN_RANGE
location: class Byte
Test.java:7: error: cannot find symbol
    System.out.println("MAXVALUE of byte is :: "+Byte.MAX_RANGE);
                                         ^
symbol: variable MAX_RANGE
location: class Byte
2 errors

D:\javap java.lang.Byte
Compiled from "Byte.java"
public final class java.lang.Byte extends java.lang.Number implements java.lang.Comparable<java.lang.Byte> {
    public static final byte MIN_VALUE;
    public static final byte MAX_VALUE;
    public static final java.lang.Class<java.lang.Byte> TYPE;
    public static final int SIZE;
    public static final int BYTES;
    public static java.lang.String toString(byte);
    public static byte valueOf(byte);
    public static byte parseByte(java.lang.String, int) throws java.lang.NumberFormatException;
    public static byte parseByte(java.lang.String) throws java.lang.NumberFormatException;
    public static java.lang.Byte valueOf(java.lang.String, int) throws java.lang.NumberFormatException;
    public static java.lang.Byte valueOf(java.lang.String) throws java.lang.NumberFormatException;
    public static java.lang.Byte decode(java.lang.String) throws java.lang.NumberFormatException;
    public java.lang.Byte(byte);
    public java.lang.Byte(JavaLangString) throws java.lang.NumberFormatException;
    public byte byteValue();
    public byte byteValue();
}
```

2 errors

Nitin Manjrekar

3

13-10-2022 (msc-project - Normal)

File Edit View

Number data

To store whole numbers we have 4 datatypes

- a. byte
- b. short
- c. int
- d. long

datatype information like

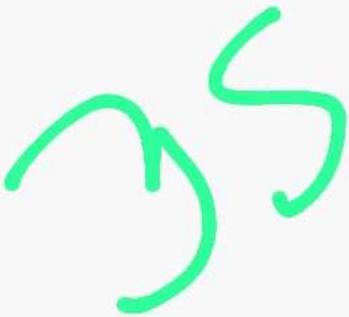
- a. size of datatype(how much memory is allocated on the ram for that datatype by JVM)
- b. minvalue what it can keep
- c. maxvalue what it can keep

Byte:

size -> 8 bits
minvalue -> -128
maxvalue -> 127

marks=35



```
*13.10.2022.class_review - Notepad  
File Edit View  
Nitin Manjrekar  
100% Windows (CEP) UTF-8  
24C Country  
L1:218 Col:3  
  
note:  
System.out.println("Size of byte is :: "+Byte.SIZE);  
System.out.println("MINVALUE of byte is :: "+Byte.MIN_VALUE);  
System.out.println("MAXVALUE of byte is :: "+Byte.MAX_VALUE);  
  
Byte:  
size-> 8 bits  
minvalue->-128  
maxvalue -> 127  
  
eg|  
byte marks=35 //valid  
byte marks = 135; //CE: possible loss of precision  
found: int  
required: byte  
  

```

Note: for boolean datatypes the only values allowed for a variable is "true,false", other than this if we try to keep any values it would result in "CompileTimeError".
=> All reserved words/names would start with "lower case".
=> In java all Classnames/interfacenames would start with "upper case".

Which of the following list contain only reservewords/keywords/builtinwords?

1. final,finally,finalize
- ans. finalize is not a reserveword, it is a method in Object class.
2. break,continue,return,exit
ans. exit is not a reserve word, it is a method in System class
3. byte,short,integer,long
ans. Integer is not a reserve word,it is a predefined class
4. throw,throws,thrown
ans. thrown is not a reserve word,it is a userdefined variable.

Datatypes

13.10.2022_Glass_Power - Notepad

File Edit View

Nitin Mehta

100% Windows (CEP) UTF-8 ENG ☰ ↻ ⌂ 13.10.2022

23°C 100% Hair cutting



13_0_2022_Class_Score - Home

10

```
byte marks = 135; //CE: possible loss of precision
byte marks = -1;//valid

byte a = true;//CE: incompatible types
byte b = "nitin";//CE: incompatible types
```

When to use byte datatype?
it is commonly used when we handle the data which is coming from stream, network.

" " -> means String data

...-> char data





```
13_10_2022_21ms_prime - Notepad  
File Edit View  
byte:  
size -> 8 bits  
minvalue -> -128  
maxvalue -> 127  
  
eg:  
byte marks=35 //valid  
byte marks = 135; //CE: possible loss of precision  
byte marks = -1;//Valid  
  
byte a = true;//CE: incompatible types  
byte b = "nitin";//CE: incompatible types  
  
When to use byte datatype?  
it is commonly used when we handle the data which is coming from stream, network.  
stream -> java.io package  
" " -> means String data  
' ' -> char data
```

When to use byte datatype?
it is commonly used when we handle the data which is coming from stream, network.

- stream -> java.io package
- " " -> means String data
- ' ' -> char data

short

```
System.out.println("Size of short is :: "+Short.SIZE);
System.out.println("MINVALUE of short is :: "+Short.MIN_VALUE);
System.out.println("MAXVALUE of short is :: "+Short.MAX_VALUE);
```

size : 16bits(2 byte)

minvalue: -32768

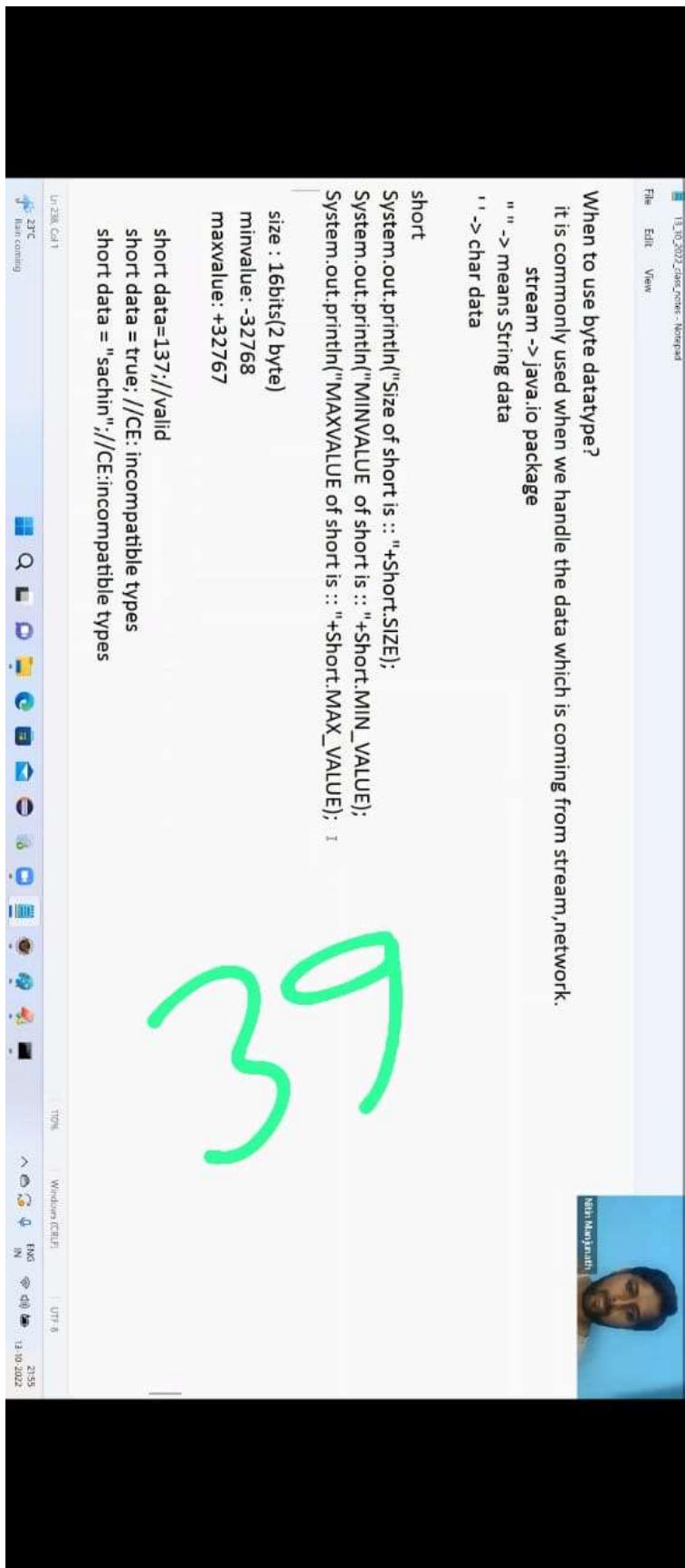
maxvalue: +32767

short data=137;//valid

short data = true; //CE: incompatible types

short data = "sachin";//CE: incompatible types

39



40

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, View, Search, Document, Project, Tools, Browser, Export, Window, and Help. The toolbar below has icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others. The left sidebar displays a file tree with Java files like Test.java and Test2.java, and a package named DemoJava. The main editor window contains the following Java code:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         short s = 1; //memory for this short data -> 16bits
6         byte b = 1; //memory for this byte data -> 8bits
7     }
8 }
9 }
10 }
11 }
```

A large green circle highlights the numbers 4 and 0 in the code. A small photo of a man, identified as Nithin Manjunath, is overlaid on the right side of the editor window.

Note: This data is not at all used in java and this data type is best suited only if u have old processors like 80

```
int:  
System.out.println("Size of int is :: "+Integer.SIZE);  
System.out.println("MINVALUE of int is :: "+Integer.MIN_VALUE);  
System.out.println("MAXVALUE of int is :: "+Integer.MAX_VALUE);  
  
size: 32bits(4 bytes)  
minvalue:-2147483648  
maxvalue: 2147483647
```

```
eg: int data = 323445;  
int result = true;//ce:incompatible types  
int result ="pass";//ce:incompatible types
```

note: The most commonly used datatype for storing whole number is "int" only and by default if we specify any literal of number type compiler will try to keep it as "int" only, but we can keep either in short or byte also.

Ln 264 Col 106
File Edit View
23°C Rainy coming



13_10_2022.java - Notepad

File Edit View

note: The most commonly used datatype for storing whole number is "int" only and by default if we specify number type compiler will try to keep it as "int" only, but we can keep either in short or byte also.

```
long  
System.out.println("Size of long is :: "+Long.SIZE);  
System.out.println("MINVALUE of long is :: "+Long.MIN_VALUE);  
System.out.println("MAXVALUE of long is :: "+Long.MAX_VALUE);  
  
size: 64bits(8bytes)  
minvalue:-9223372036854775807  
maxvalue:9223372036854775807  
  
e.g. long data = 10;
```

A2

Note:
When int is not enough to hold the big values, then we use long datatype.
When we work with large files, data would come to java program in terms of GB's.

```
long size = file.length();
```

13_10_2022.java - Notepad

File Edit View

note: The most commonly used datatype for storing whole number is "int" only and by default if we specify number type compiler will try to keep it as "int" only, but we can keep either in short or byte also.

```
long  
System.out.println("Size of long is :: "+Long.SIZE);  
System.out.println("MINVALUE of long is :: "+Long.MIN_VALUE);  
System.out.println("MAXVALUE of long is :: "+Long.MAX_VALUE);  
  
size: 64bits(8bytes)  
minvalue:-9223372036854775807  
maxvalue:9223372036854775807  
  
e.g. long data = 10;
```

A2

Note:
When int is not enough to hold the big values, then we use long datatype.
When we work with large files, data would come to java program in terms of GB's.

```
long size = file.length();
```

File Edit View

eg: long data = 10;
long data = 9223372036854775807;

If the data goes beyond the range of int, then to keep the data inside data with 'L' or 'l' otherwise it would result in "CompileTimeError".

e.g. long instudata=3223372036854775807;/;CE
long secodData=9223372036854775807;/;CE

```
long firstData=9223372036854775807L;  
long secodData=9223372036854775807L;
```

```
long data = 10L;  
long number = 5L;
```

Note:

When int is not enough to hold the big values, then we use long data type.

```
long size = file.length();
```