java -version : → whether java is installed or not

javac -version
↳ x

JDK
↳
JRE
JVM
javac

D:// Learn java
↳ 1.java
javac ← ↳ Compile → .class (Bytecode)
↳ execute / Interpret

Env. variable to configure javac

→ Edit System Env. Variable.

→ Env. Variables < user → Specific user

System → for All the System Users.

→ path → Edit → NEW → paste the path of JDK folder

C:\Program Files\Java\jdk1.8.0_271\bin

→ OK → OK → OK

→ Open New cmd and run command

javac -version → javac 1.8

Notepad ++] →

Notepad

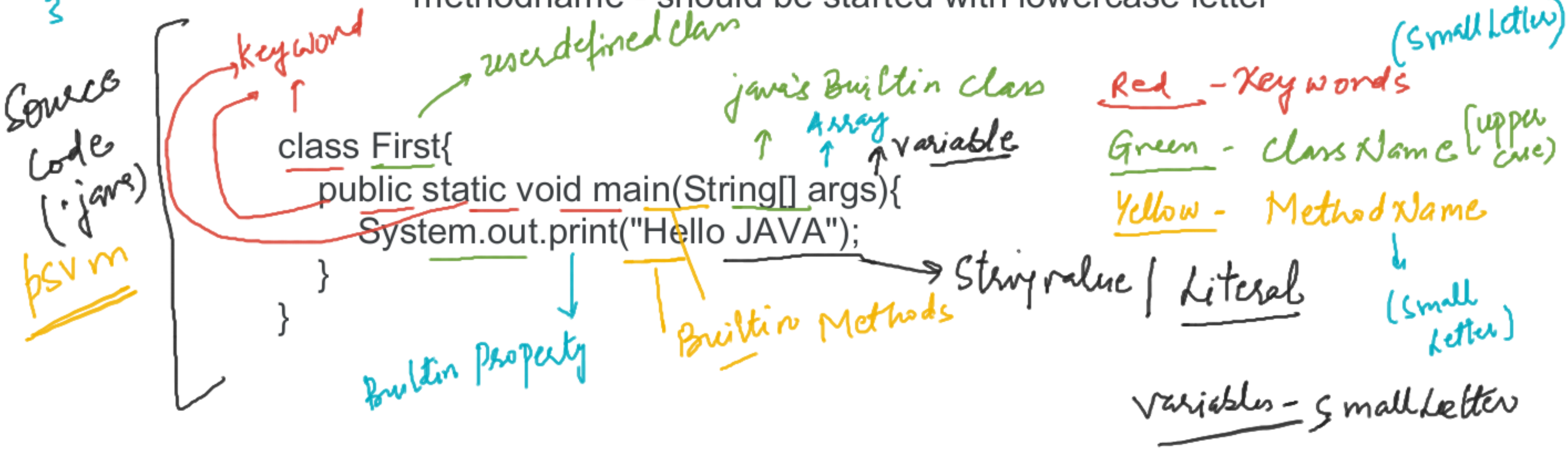
Name Convention: relevant / proper name

Creating any class, fileName,methodName

Classname - First letter of class Name should be in uppercase and
File name: is also starts with upplercase letter

{ block }

methodname - should be started with lowercase letter



(i) className and FileName is different – Yes, it compiles & executes the code successfully.

(ii) className & Filename is Same → Yes, "

(iii) If we write 'public' keyword infront of class, incase className and FileName is diff. → Error at compile-time.

We need to keep the file & className same, if we are using public keyword

Program Execution Flow

1. Source Code → Test.java

compile → [javac Test.java] ①

filename

ByteCode → First.class

2. Execute/Interpret the ByteCode → JVM executes it

[java First] ②

↓ className

JVM starts the execution from main() method.

```
public class Test{
    public static void main(String[] args){
        System.out.print("Hello JAVA");
    }
}
```

→ Creating a class

→ main method from where JVM starts the execution

↓

To print the statement

int a = 10

a++ ✓ post increment

+1

Syso(a); // 11

++a; // +1 = 12

pre increment

a++ 13

Syso(a);

int a = 12

a++; // 13

++a; //14

a--;

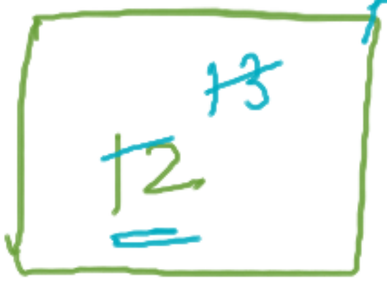
--a;

a++;

Syso(a); // 13

M|m

M|m



int x = 1;

Syso(x++);

1 2

Post increment → 2 Action

Action-1 → First value

Syso(x); → 2

int y = 2;

y++; // 3

++y; // 4

Syso(y++); // 4

Syso(++y); // 6

syso(y); // 6

M|m 6 5

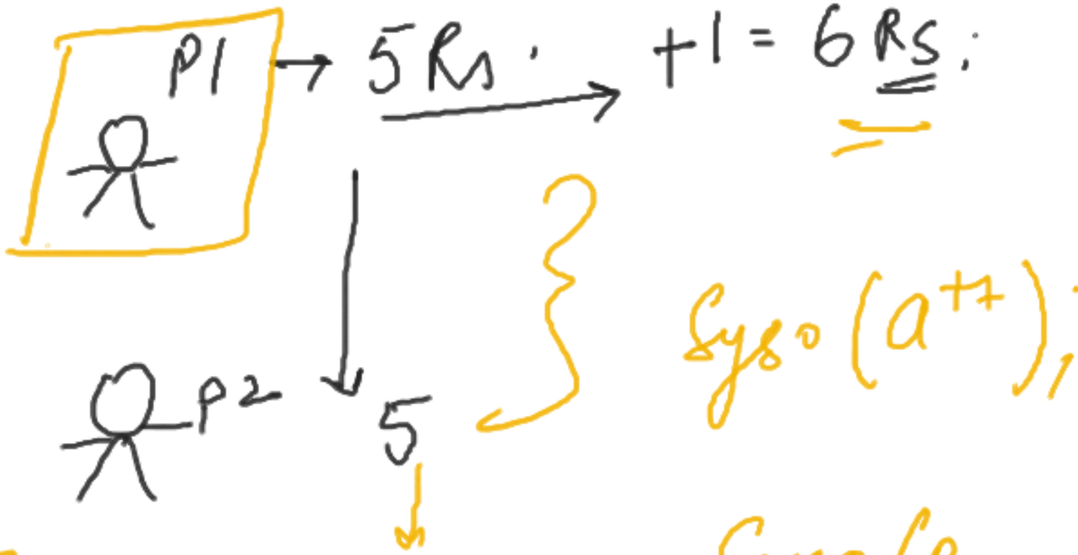
2 3 4

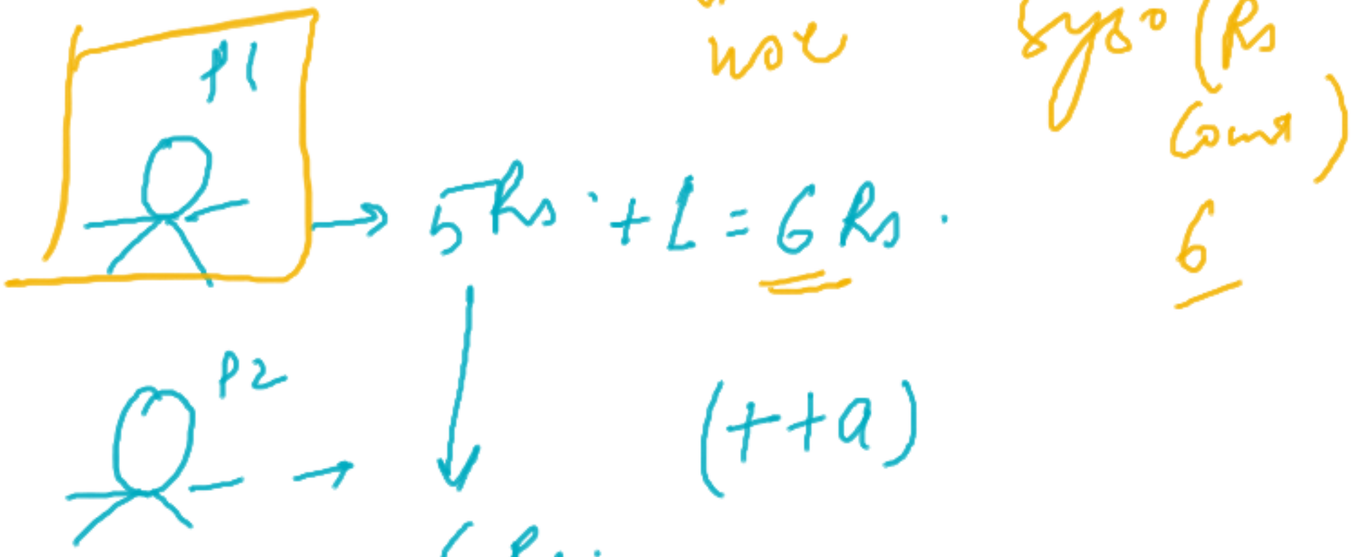
y

Pre-increment

```
int a = 2;
Syso(++a);
```

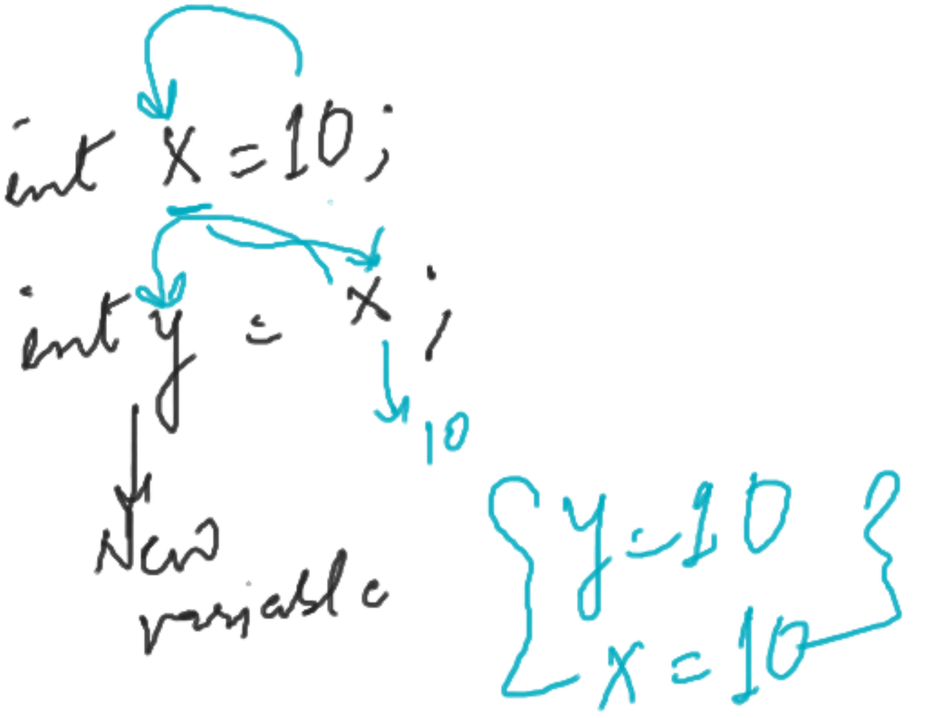
(2) (1) → 3





Ex- pre-post increment

```
int x = 10;
    x++; // 11
    ++x; // 12
Syso(x++); //12
Syso(++x); //14
    Syso(x); //14
```



declaration → int x, int y

init → (x)

(11) assign the value of pre-post increment in a variable.

```
int x=10;
int y=x++;
Syso(x); //
Syso(y); //
```

NV

y

→ op1 → value assign → y = 10

→ op2 → value increment

Data Types

JVM
↓
(interpreter)

→ primitive DT

→ Non primitive DT

that can store a single value in a variable

that can hold multiple values in a variable.

1 Bit → 1 0 1 0

Bit

1 Byte → 8 bits

→ primitive DT

byte
short
int
long
} numeric integers

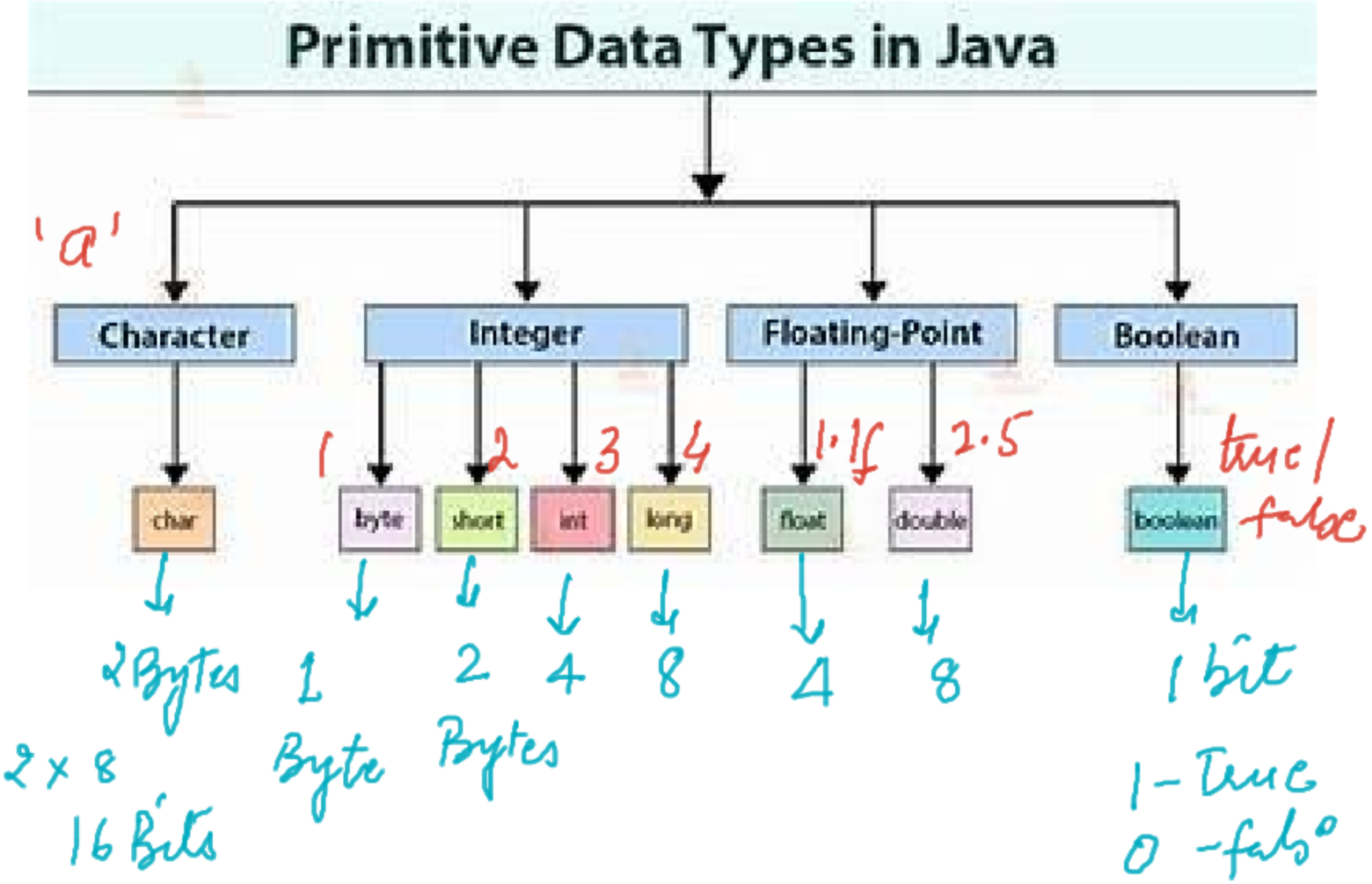
float
double
} decimal (.)

boolean → true, false
char → single char
a, 5, c

eg:

Binary lang

128	64	32	16	8	4	2	1
1 →				0	0	0	1
2 →				0	0	1	0
3 →			0	0	1	1	0



byte b = 10;

0 0 0 0 1 0 1 0

Bit 128 64 32 16 8 4 2 1

Primitive Data Types - Summary

Data Type	Keyword	Size(bit)	Size(byte)	Min	Max
character	char	16	2	0	65535
byte	byte	8	1	-128	127
short	short	16	2	-32768	32767
integer	int	32	4	-2e31	-2e31 -1
float	float	32	4	1.4e-45	3.4e+38
long	long	64	8	-2e63	-2e63-1
double	double	64	8	4.9e-324	1.8e+308
boolean	boolean	NA	NA	false	true

1 Byte

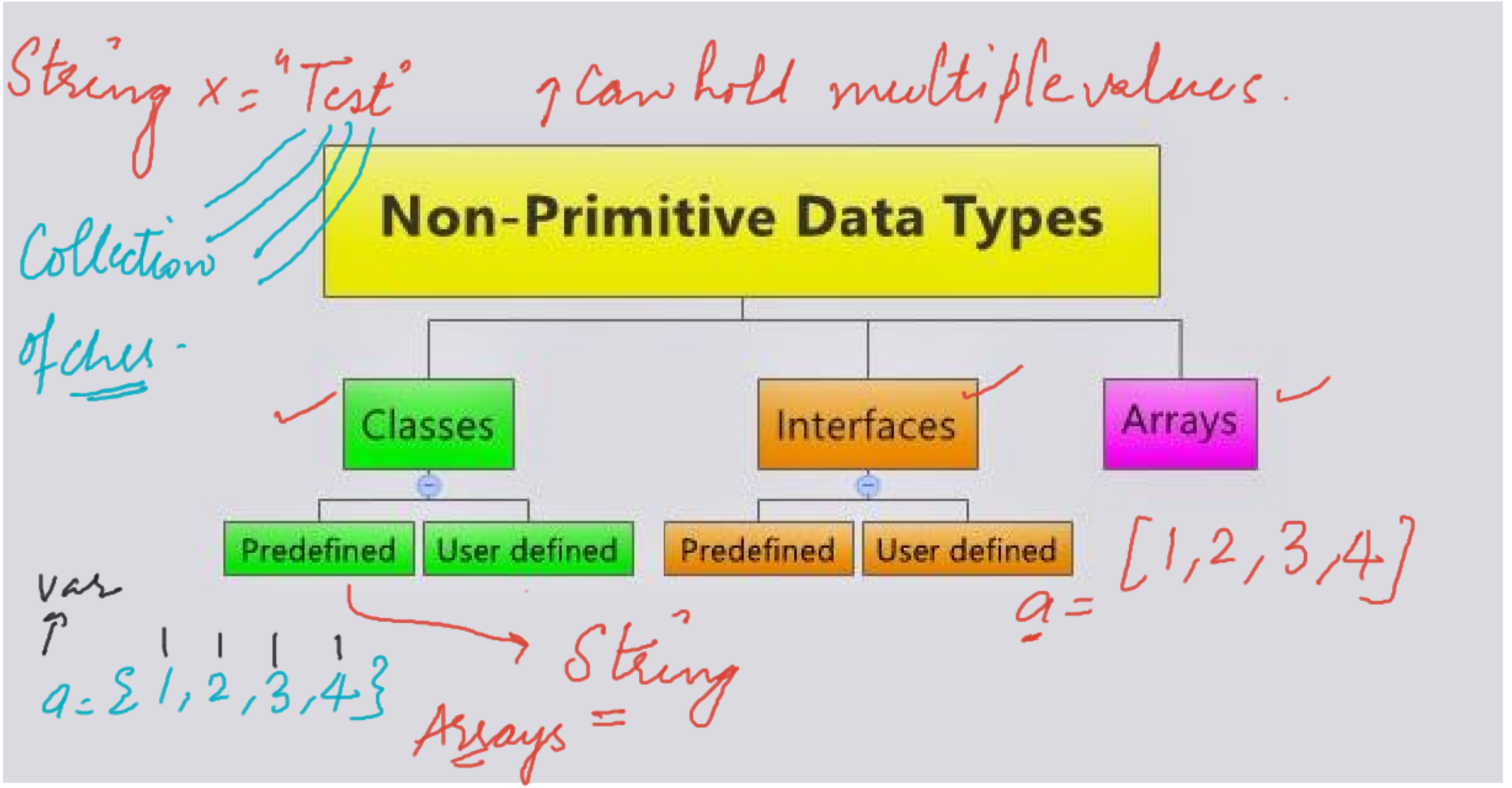
x 8

x 8

(II) Non-primitive Data Type:

String x = "Test" → can hold multiple values.

Collection of chars.



var

int[] a = {1, 2, 3, 4}

String

Arrays =

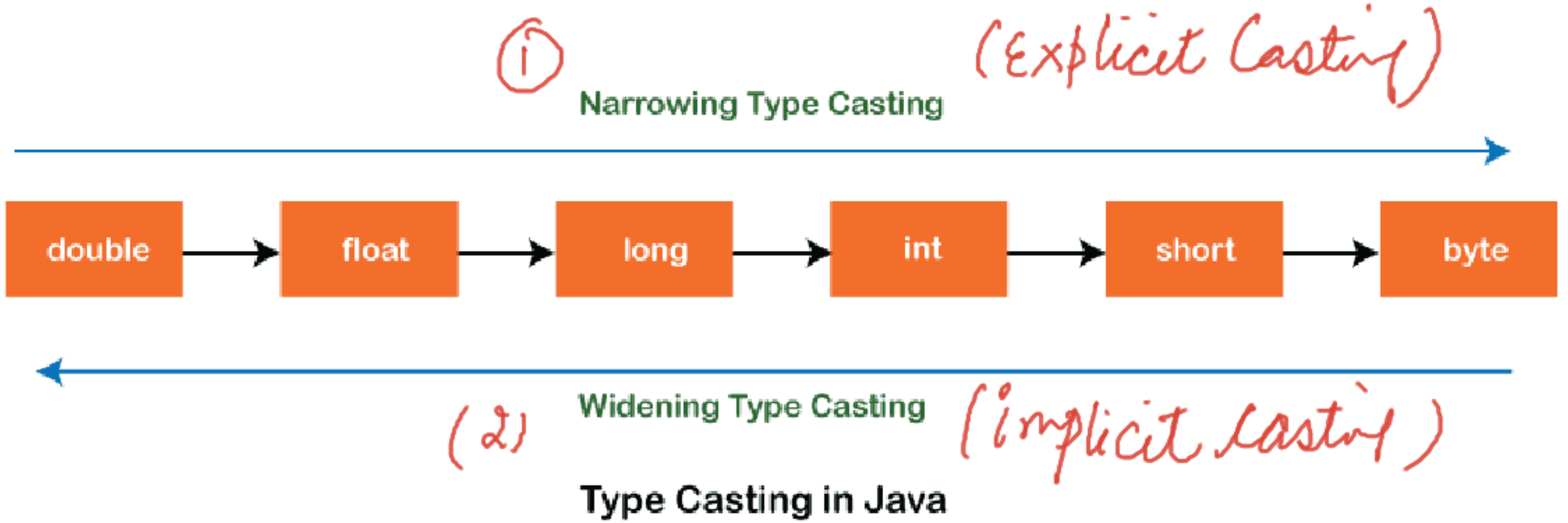
a = [1, 2, 3, 4]

Type Casting :-

Star Cast

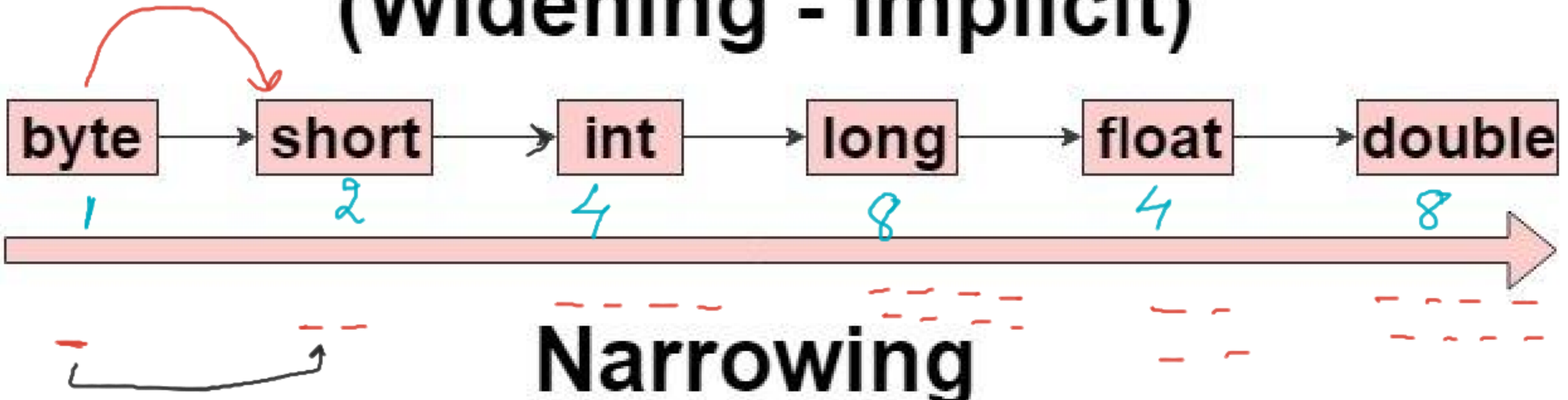
→ Role

Conversion/ storing the value of one datatype into another datatype.



Type Casting in Java

Automatic Type Conversion (Widening - implicit)



Narrowing (explicit)