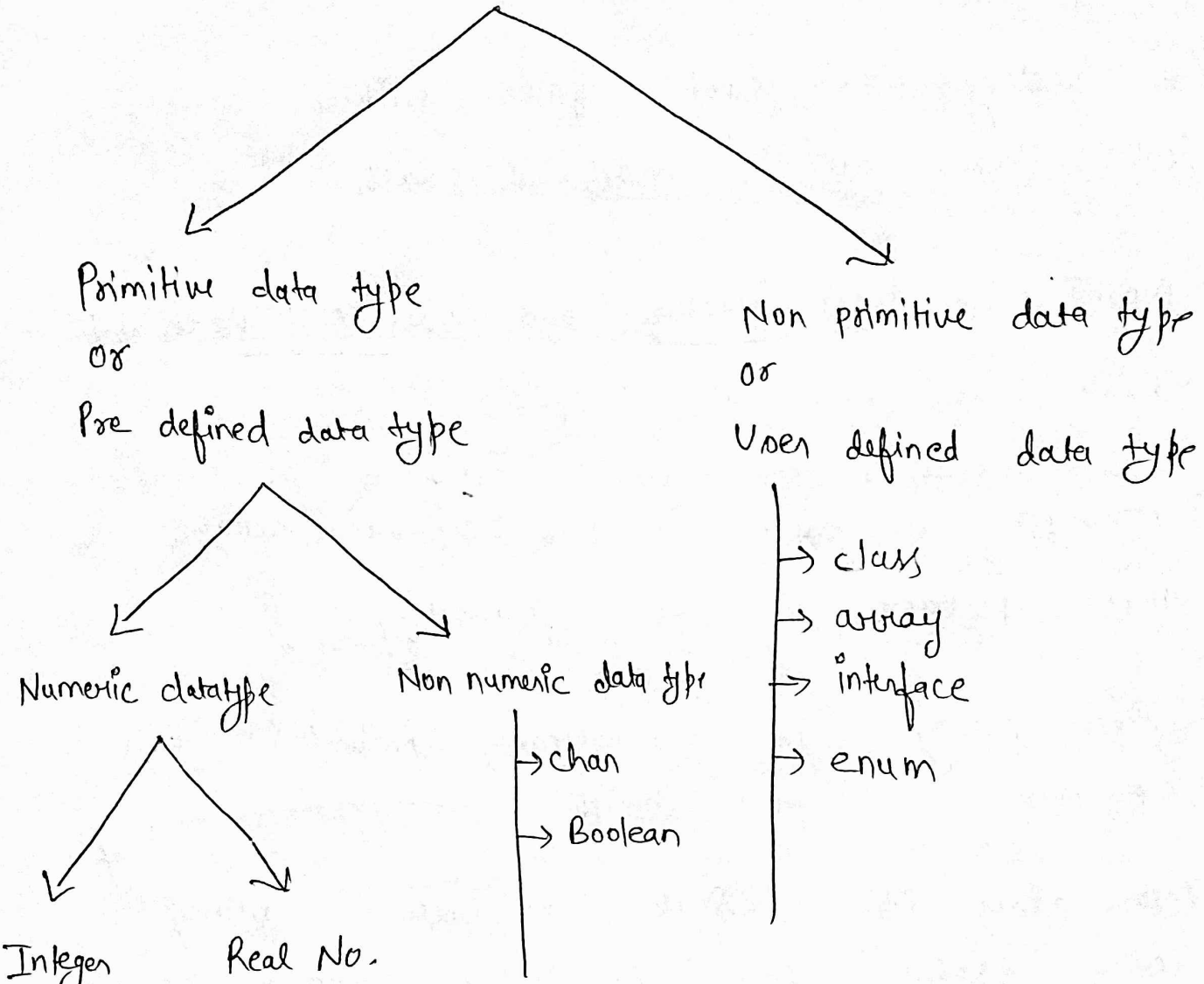
Data type


Primitive data type
or
Pre defined data type
Non primitive data type
or
User defined data type
Numeric datatype
Non numeric data type
Integer
Real No.
(float)
→ Byte
→ Short
→ int
→ long
→ float
→ double
→ char
→ Boolean
→ class
→ array
→ interface
→ enum

Data type	Size (In byte)	Range
① Byte	$1B \Rightarrow (2^0)$	$-128 = 127$ $-(2^7-1)$ to (2^7-1)
② Short	$2B \Rightarrow (2^1)$	$-32768 — 32767$ $-(2^{15}-1)$ to $(2^{15}-1)$
③ int	$4B \Rightarrow (2^2)$	$-2147483648 —$ 2147483647
④ long	$8B \Rightarrow (2^3)$	$-9,223,372,036,854,775,$ $808 — 9,223,372,$ $036,854,807.$
⑤ float	4B	-3.4×10^{38} to 34×10^{38}
⑥ double	8B	-1.7×10^{308} to 1.7×10^{308}
⑦ Char	2B	$0 - 65535$
⑧ Boolean	JVM (depend on) ↙ ↘ 1B 1bit	True, false

☆ In Java ASCII → X
UNICODE → ✓ ⤵ in place of

***

In java UNICODE happen bcz in ASCII only support english lang. but UNICODE support 61 spoken language.

UNIcode :-

'A' → 65

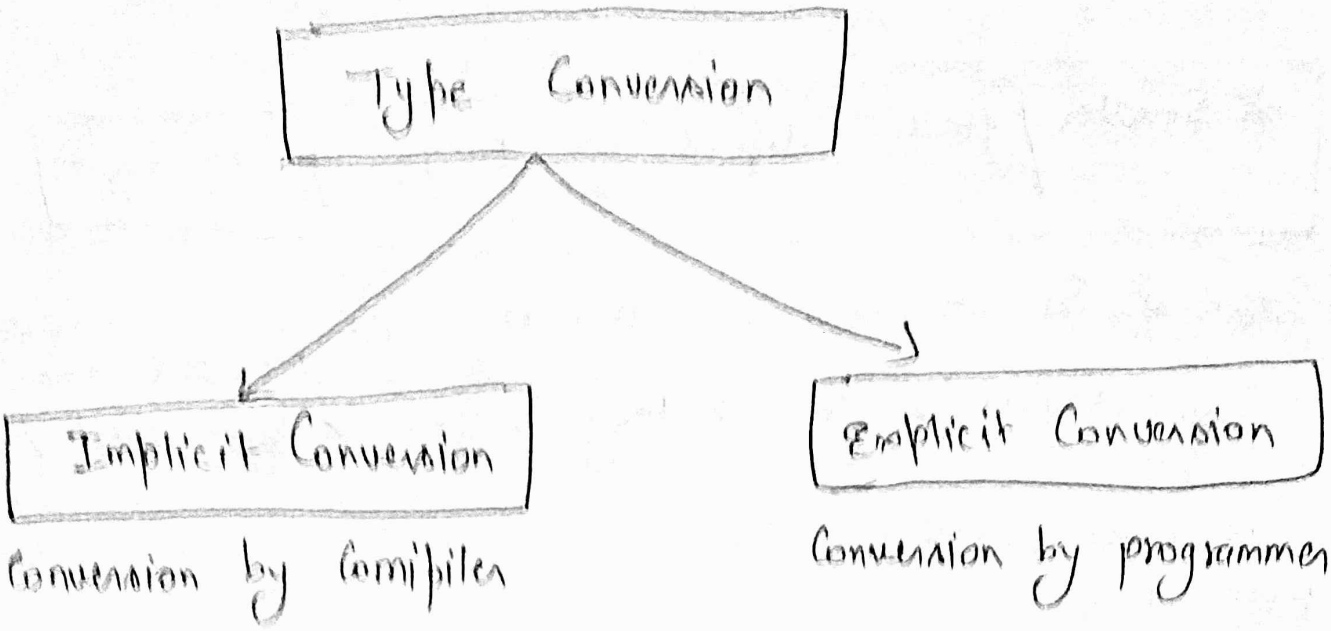
'B' → 66

'a' → 97

'z' → 122

} also this ^ work ~~also~~ in java in unicode.

Type Conversion :-



There are two Rule when Implicit Conversion happen

(1) Values on both Side of assignment operator must be Compatible

Ex.

```
int n;
n = 'A';
print(n);
```

→ unicode

65 → Compiler Conversion करेगा

Unicode → 'A' → 65

```
int n;
n = true
print(n);
```

→ Boolean

Syntax Error → programmer को Conversion करना पड़ेगा।

(2) The value of R.H.S of assignment operator must be Smaller than the Variable on the left of assignment.

> ✱ Smaller / greater depend on Range of data type

✱✱✱ छोटा बड़े में जा सकता है। लेकिन बड़ा छोटे में नहीं जाएगा generate कर देगा error.

✱ double > float > long > int > ~~byte~~ Short > byte

Ex.

```
float a;
a = 10;
print (a)
```

→ Compiler will perform type Conversion.

10.000000

Ex.

```
int a;
a = 1.5   → double
print (a)
```

Syntax error

programmer convert / solution

Rule break

```
int a;
a = (int) 1.5
print (a)
```

1

Some Codes :-

(i) byte a = 10 ;
int b ;
b = a ;

✓ it works (Run)

2) int a = 10 ;
byte b ;
b = a

✗ Not work (Second Rule ~~break~~)

*** very important point :-

* Byte छोटा है or int बड़ा है, So
** छोटा बड़े के अन्दर Run हो जाएगा

3) int a = 128 ;
byte b ;
b = a ;

✗ Not Run.

☆

छोटा → byte
Short
Int
long
float
सबसे बड़ा → double

Solution of previous code :-

```
int a = 10;
byte b ;  → X Not Run.
b = a ;
```

Solution

```
int a = 10
byte b;
b = (byte) a  → user change data type
                Explicit data Conversion.
```

Some other example

#
```
Short a = 10;
int b;  → ✓ Run (Becz Short छोटा है int बड़ा है तो छोटा बड़े में चले जायेगा)
b = a;
```

2) `int n = 10;`

`short s;` ✗ Not Run

`s = n`

short ← int

✗ नहीं जा सकता है।

`int a = 10;`

`long b;`

`b = a;` Run ✓

long ← int

✓

`long a = 10;`

`int b;` Not Run.

`b = a;`

int ← long

✗

Very imp.

to Solution → `long a = 2147483648L`

↓

this will work

`long a = 2147483648;`

`int b;`

`b = a` ✗

2147483648 फिर,

Number treated as integer or integer Range is less

So Java इस Number को

Integer Number too large मान रही है।

long very imp point!-

कभी भी long Number देंगे आप तो last में L लगाना जरूरी है otherwise java treated as integer or integer की Range होती होगा है तो java उस No. को कुछ नहीं मानेगी or generate कर देगी error

Ex long a = 10 → treated as int

but

long a = 10L → treated as long ✓

⭐ long में होना use Suffix L

⭐ If we use . (dot) → this Convert to double