01/11/22

Data-types:- Every Variable has a type, Every Expression has a type and all types are strictly typed / defined in java because java is strict type / statically type language.

$$\begin{tikzcd} \text{int data = 10;} \arrow[r, "data"] & \text{Variable} \end{tikzcd}$$

(i) int data = 10;

```
int a = 10;
int b = 20;
int Result = a*b;
```

$\rightarrow$ Expression

→ Compiler Role:- Compiler ~~checks~~ will check the Value Stored Can be handled by data type Can not.

→ This Checking which is done by Compiler is Called "TypeChecking / strict type checking".

```
int data = 10;        → keyword
~~boolean data = true;~~
boolean result = true;
```

→ Compiler will check whether this 10 Can be stored in data Can not.

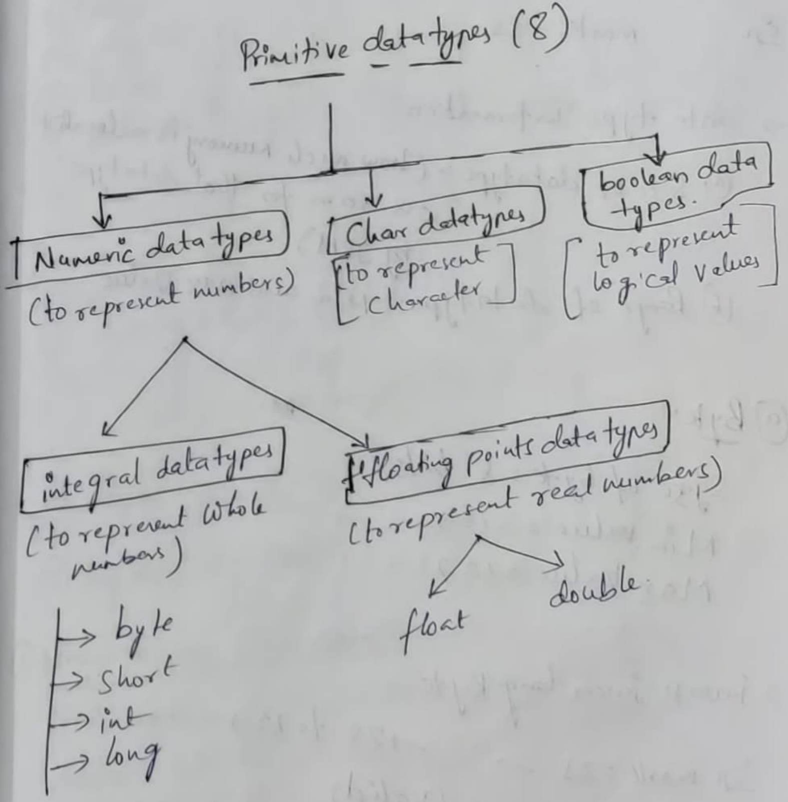
→ Java strit type language

① Primitive data types:- (data which is Commonly used and supported by language to Store directly.

ⓐ Numeric Values :- ~~a~~ to store number. → a. whole number → b. real number

ⓑ Character Values:- to store character type of data

ⓒ boolean values :- to store logical Values.



① Numeric data :- to store whole numbers.

→ we have foure data types

ⓐ byte
ⓑ Short
ⓒ int
ⓓ long.

Ex marks = 35

→ data type Information.

ⓐ Size of datatype :- (how much memory is allocated on ram for that datatype by JVM)

ⓑ Range of datatype :- Min and max Value.

ⓐ Byte :-

Size of byte = 8 bits
Min value = -128
Max Value = 127

→ javap java.lang.Byte

Ex mark = 35 → -128 to 127

byte mark = 35 (Valid)
byte Mark = 135 ; // CE (Possible loss of Precision)
found int type
required : byte.

→ Into already known to Compiler (Reserved Words)

Reserved Words for datatypes (8) :-

① byte
② Short
③ int
④ long
⑤ float
⑥ double
⑦ Char
⑧ boolean

Ex

① byte a = true; (invalid)
↳ C.E
↳ Incompatable Error

② byte a = 10 (Valid)

③ byte a = "nitin"; → C.E
↳ Incompatable types.

① When to use byte data type?

it is Commonly used when we handle the data which is Comming from stream network.

→ Stream → Java.io. package.
① " " → String data ② ' ' → Char data.

Note:- ① all reserved words names would start with "lower Case".
② In Java all Class names/interface names would start with "upper Case".

④ Code to know Size, Min and Max Value of D.T

```
System.out.print("Size of Byte" + Byte.SIZE)
System.out.println("Min Value" + Byte.MIN_VALUE);
System.out.println("Max Value" + Byte.MAX_VALUE);
```

② Short data type:-

1 byte = 8 bits
2 = x
~~16~~

Size of Short = 16 bits (2 byte)
Min value = -32768
Max Value = 32767

Eg: short data = 137; → valid
Short data = true; → C.E
↳ Incompatible types

short data = "Sachin" → C.E
↳ Incompatible types.

Short s = 1; // Memory = 16 bits
byte b = 1; Memory = 8 bits.

→ for Memory utilization we can use large data type.

Note:- this data is not at all used in java and this data type is best suited only if you have old processors like 8086.

→ int and float are Commonly used data types.

③ int data types:-

Size of int is = 32 bits (4 bytes)
Min Value of int = -2147483648
Max Value of int = 2147483647.

Eg: int data = 323445; (valid)
int result = true; //CE : inCompatible types
int result = "Pass"; //CE; "

Note:- The Most Commonly used data type 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.
→ we Can also keep Either in short or byte also.

④ Long Data type :- · 1 byte = 8 bits.

size of long = 64 bits (8 bytes)

Minvalue = -9223372036854775808

Max Value = 9223372036854775807

→ when we work with large files, data would come to java program in terms of GB's

→ when int is not enough to hold the bigger values, then we use long data-types.

$\boxed{\text{long size = file.length();}}$

Eg :- long data = 10;

long data = 9223372036854775807~~1~~; → L or l

↳ ~~[illegible]~~ it is reading as int.

↳ so we have give L ~~to~~ to treat as long.

→ if the data goes beyond the range of int, then to keep the data inside long data-type we need to Explicity suffix the data with 'L' (or) 'l' otherwise it would result in Compiler Time Error.

Eg ① long a = 9223372036854775807L; (valid)

② long b = 9223372036854775807l; (Valid)

→ long c = 10L (valid)

Examples :-

Case ①

Eg byte a = 10;

byte b = 5;

byte result = a*b;

System.out.println(result);

~~S.o.P~~ a

→ Possible lossy Conversions from int to byte.

↳ Compile time Error.

Case ②

byte a = 10;

byte b = 5;

byte result = a+b;

System.out.println(result);

System.out.println(a+b)

doubt Session :

① As per requirement you can manage the type of data is good (or no management universally will keep. is it good?

operations on → N/w → bytes → byte data. (1 byte occupies)

doing → normal calculation → integer - int data (4 bytes)

on → files whose size is big → long → long data (8 bytes)

(or
→ Every operation → long data → long