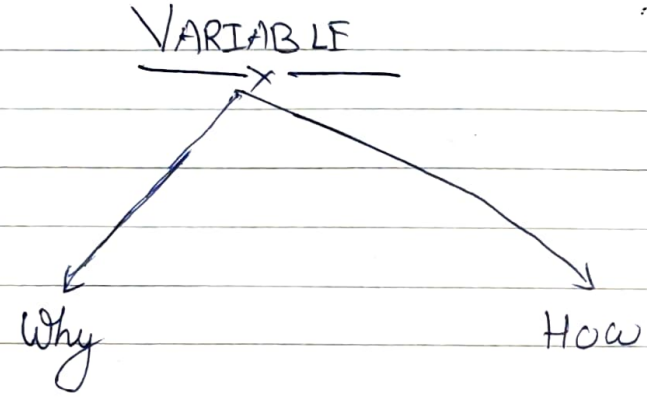
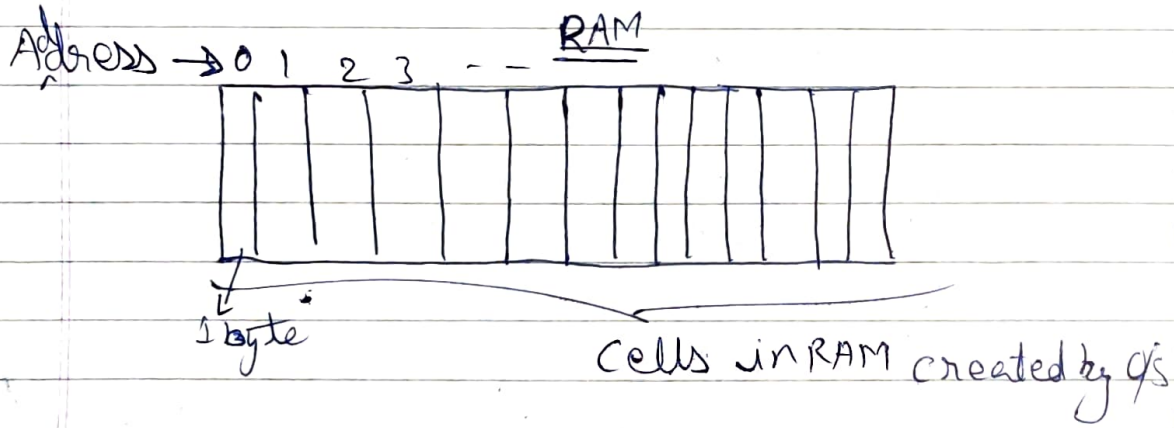
'public static....' & another by 'static public..' On executing the program will give error saying two 'same main' is being used

14/06/17





Program reserves some of space of RAM This is known as Variable.

Variable are just a name given to a memory location. & that that memory location is used to hold the data

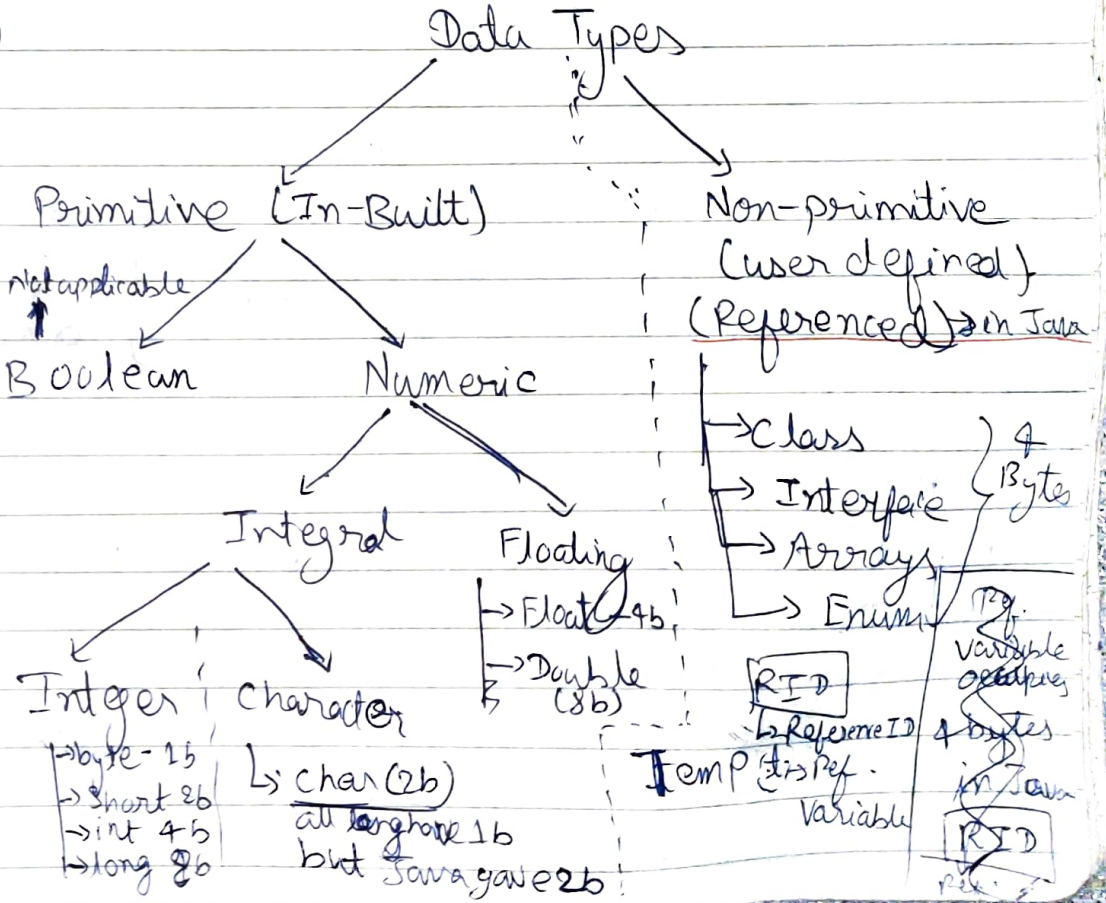
Note: We can ~~be~~ also go to that particular ~~area~~ cells, by their addresses, this is known as pointers.

- JAVA Don't have Pointers because of their complexity (P ") and ~~dont~~ danger of data leaking / breaching.

DATA TYPES

Constructed for proper reservation of cells in RAM & provide mutual ~~or~~ understanding b/w user & prog lng for proper memory allocation.

(i)



Since Boolean only has constants either 'True' / 'False' so it points to 'NULL'. So, no memory is given to Boolean. It just checks with Logical & Operator.

15/06/17

- Concept for

1 byte = 8 bits

+

Latin → 256

We alloted binary code for every 2^8 Latin Characters.

And to accomodate every Latin Character in binary we needed 2^8 Combinations ($256 = 2^8$)

for this 8 bits are required as

1	2	3	4	5	6	7	8

From here 1 byte = 8 bit

Now,

A No. is alloted to every Latin No. from

0, 1, 2 - - - - - 255

This is known as ASCII.

* Now, → There were other languages too
A new code system is made in which following code system is used.

0–255 , 256 , 257 - - -
Latin Other Lang.

For this '2 Byte' code system is used. → in a 16 bit machine
$2^{16} - 1 = 65535$

• Now, → Above system fell shor so Unicode system is used in which every No. is instead of dividing by (2) we divided by '16' 4 stores the [value → To convert to binary we divide by (2)
So, More than 65535 can be used. for same 16 bit machine

★ Java Supports UNICODE SYSTEM

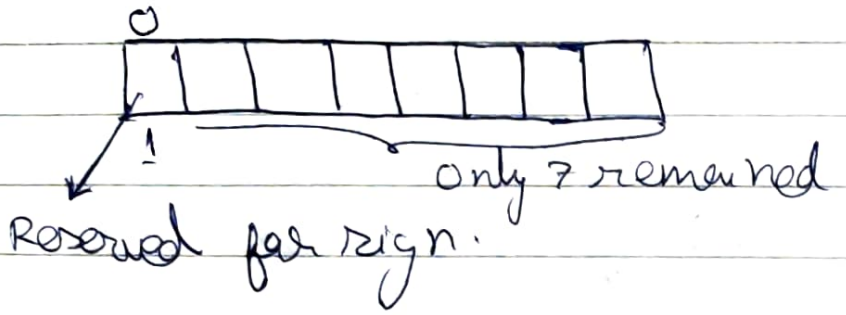
✓ In Java all the Numeric datatypes are 'Signed'

• If you define any variable as a Signed that means that variable can hold both '+ve' & '-ve' values

• If you define any Variable as a unsigned that means that variable can hold only '+ve' values.

In case of signed

lets byte b = -10;



-2^{n-1} to $2^{n-1}-1$; n = total no. of bits

For holding zero

-2^{8-1} to $2^{8-1}-1$

-128 to 127

- In floating type / Double some bits are reserved for whole No. others are reserved for fractions. This follows some predefined rules.

- In Java every data type has got a default value. (This rule belongs to class level variables.)
  Means there is no Garbage value in any datatype in Java.

byte = 0 (zero)	float = 0.0f
short = 0	double = 0.0d → optional (only here)
int = 0	boolean = false
long = 0L	Char = /u0000 (these are zeros)
	Ref. Variable = null

- /0 → represents NULL in ASCII
- /u0000 → Represents null in Unicode (smallest Value)
- /uffff → " (biggest value in Unicode)
  (16×16×16×16)
- 16 | 65 | 4 → 65 → /u0041
  64
  1
- In Java there is no NULL (capital); but there is null (small)
- In Java all variables are also defined within class
  There is no concept of Global Variable.

P.T.O

Evergreen
Page No.
Date: / /201

```
Class Temp.
{
int x;
int y;
```
} class Level Variables

```
psvm( )
{
 int a;
 int b;
}}
```
} Local Variables

Imp. Local Variables must be initialised before the first use.

Eg.

```
Class Temp
{
psvm( )
  {
   int x;
   int y;
   int z = x + y;
   Sop(z);
  }
}
```
} Compilation error.

↳ Here single quotes gives us output → z
doubles " " " → z
no " " " → 60
as when

if we define x = 4 to before z = x + y program runs.

Evergreen
Page No.
Date: / /201

If — How Why — byte b = 10 (byte missing) — it will bind to byte ??

Show (b)

prog. will bind 'b' with proper data type. i.e. here it will bind with 'byte'

void Show (byte b)
" " (short s)
" " (int x)
" " (long l)

?? → How

Now,

if

show (10) → a literal/constant
→ it will belong to 'int' in java.

Java

- In Java all integer literals are treated as a 'int.'

But some times, if
Show (10000000000)
outside the range of 'int'.

We can use a suffix 'L' and save it from data route ??
such as
Show (100000000L)
& pass it to 'long'.

- Only applicable to 'long' not to 'byte' or anything.

IQ →

```
Class Temp
psvm()
{
byte b=10;
,, c=20;
,, d=b+c;
sop(d);
}
```

Ans → Its a compilation error.
No, the ans. will not be '30'.

Old Prog. lang { The result after operation are always literals and we can catch it in any data type.

JAVA { But in Java, all literals belong to 'int' in prog. we tried to catch '30' in 'byte' hence leads to compilation error.

For solving error we use typecasting

```
byte d = (byte)(b+c)
sop(d);
```

Now, in floating Literal

show(2.4);

void show(float f)

~~void show(float f)~~

void show(double d)

- In Java all the floating literals are treated as 'double'
- If we want to treat as float we use it as
  ↳ show(2.4f);
  (↳ suffix.

~~(Java ki Gazab Kahani)~~

AJAB JAVA KI GAZAB KAHANI

- byte b = 10; → it should be a error but is not a error it is defined from intial as 'byte'
- but if
  float z = 2.4; → it gives error.
  to store as float we use suffix 'f'

```
float z = 2.4f;
```

16/06/2017

TYPE CASTING

The process of converting the value of 1 datatype into another data type is known as type casting.

- When oper^n is performed b/w 2 similar datatypes; the resultant is also in same datatype.

eg.

```
int M = 10;
int f = 3;
float z = (float)M / f;
```

↓ Typecasted only during op^n.

Types of Typecasting

(i) Implicit → Bz Smaller to bigger datatype

eg.

```
byte b = 10;
int x = b;
```

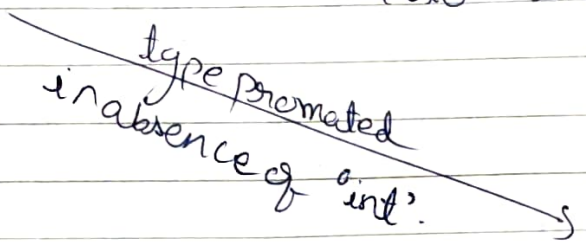
(ii) Explicit →

eg

```
int x = 10;
byte b = (byte)x;
```

- Java introduced concept of Type Promotion in Implicit Typecasting - Promotion of variable / constant to upper datatype in absence of default datatype.

show(10);



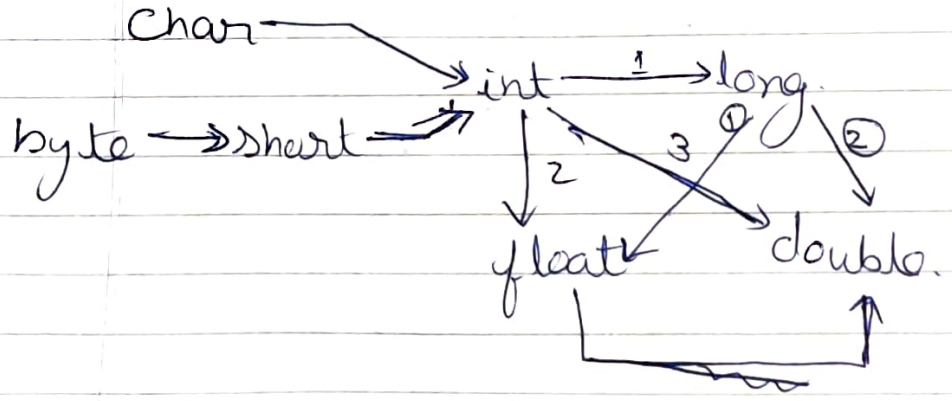
void show (byte b)

(short s)

(long l)

It doesn't do Type Demotion.
means if also 'long' is absent then there will be runtime error.

Chart for Type Promotion



here 1,2,3 indicate priority

Date / /
Page

Q → long → 8byte ; float → 4byte ; Is there a chance of data loss?

Ans → Yes, but there is also a very minute chance. as converting from long to float it converts to power of 10.

eg 1234 → 1.234×10^3

but if there are

Bad

OPERATOR

1. Precedence

Symbol which perform calculation

(i) Unary → Needs single operand → on which opⁿ is performed

(ii) Binary → " 2 "

(iii) Tertiary → " 3 "