6 Variables and data structure

* **Type Conversion [or type casting]** — type → type

conversion from one type to another

Conversion happens when:

a) type compatable
b) destination type > source type

byte → short → int → float → long → double

This is known as widening conver., implicit (automatically) conversion or type conversion. (smaller to larger)

If we try to convert the smaller to larger data type, java allows but vice versa can be only possible with some method

```
int a = sc.nextfloat();
System.out.println(a)
```

Output → It will show an error

but,

```
Float a = sc.nextfloat();
System.out.println(a)
```

User Input = 16

Output = 16.0 [int ko float bana diya]

4 bytes
25 int

8 bytes
25 long

```
int a = 25;
long b = a;
```

long b = a;

8 bytes > 4 bytes

destination type > source type

This is known as implicit, widening or type conversion in Java.

Type Casting

- larger to smaller
- data loss happens
- lossy conversion

- Java do not allow this conversion automatically.
- We do this conversion forcefully.
- Also known as Narrowing / Explicit Conversion.

eg

```
float a = 25.12f;
int b = (int) a;        // double b = (float) b
System.out.println(b);
```

↓ Jisme convert karna hai

Output → 25

```
char ch = 'a'
char ch2 = 'b'
int num = ch;        // c → 99, d → 100
int num2 = ch2;
System.out.println(num);
System.out.println(num2);
```

Output :- 97
98

Type Promotion in Expression

eg `a+b*c/e`

int float char long

① Java promotes each "byte", "short", or "char" to "int".

byte/short/char → int

i)
```
char a = 'a'
char b = 'b'
System.out.println(b-a);
```
Output = 1

Type promotion only can happen in expression

ii)
```
short a = 5;
byte b = 25;
char c = 'c'
byte bt = a+b+c;
```
"we can remove error by using type casting"
```
byte bt = (byte)(a+b+c);
sout(bt);
```

Output: error: from int to byte is not possible
(Here from rule 1 java convert all three value into int)

② If one operand is long, float or double the whole expression is promoted to long, float, or double respectively.

Meaning → There are 3/4 type of data type is given so in that case whole expression will convert into larger data type.

i)
```
int a = 10;
float b = 20.25f
long c = 25;
double d = 30;
double ans = a+b+c+d; // from rule 2
sout(ans);
```
here, java had converted the a, b, c, d in double by itself so if we try to assign int it will throw error

Rule ①

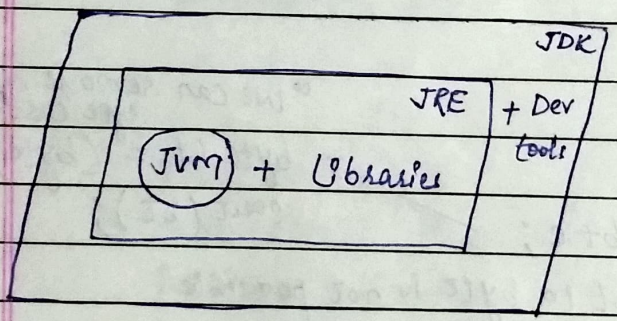
byte
short → int
char

Given

Rule ②

long
float → double (converted into greater data)
double

* How is our code Running



Java is portable because Java's byte code .class (after comp. of source code .java) can run on any other system.

C++ is not portable like Java.

② Execution stage

Source code → Compiler → Byte Code → JVM → Native code
.Java .class

① Compilation Stage

Machine Code (0, 1)