TYPE CAST OPERATOR

There are two type of type-casting

1. Implicit Type-casting

2. Explicit Type-casting

**Implicit Type Casting**

1. Compiler is responsible to perform implicit type casting

2. Whenever we are assigning smaller data-type value to bigger data-type variable implicit type-casting will be performed.

3. It is also known as widening or upcasting

4. There is no loss of information in this type casting

5. The following are variable possible conversion where implicit type casting will be perform

byte -> short -> int -> long -> float -> double

|

char

Example:

**int** x = **'a'**; *//compiler converts char to int automatically by implicit type casting*

System.***out***.println(x);

**double** d = 10; *//compiler converts int to double automatically by implicit type casting*

System.***out***.println(d);

Output:

97

10.0

**Explicit Type casting:**

1. Programmer is responsible to perform explicit type-casting

2. Whenever we are assigning bigger data-type value to smaller data-type variable then explicit type casting will be required.

3. It is also known as Narrowoing or downcasting

4. There may be chance of loss of information in-this type casting.

byte <- short <- int <- long <- float <- double

|

char

The following are various possibilities where explicity type-casting is required.

**int** x = 130;

**byte** b = x; //java: incompatible types: possible lossy conversion from int to byte

System.***out***.println(b); //-126

Whenever we are assigning bigger datatype value to smaller datatype variable by explicit type casting the most significant bits will be lost we have to consider only we have to consider only least significant bits.

int x = 130 ==> 0000 ... 010000010

byte b = (byte) x ==> 10000010

last bit is signed so the representation is in 2’s complement

1111101

1

-----------

1111110

which is 126, with last signed bit it is minus 126

If we assign floating point values to the integral types by explicit type casting the digits after that decimal point will be lost.

double d = 130.456;

int x = (int) d;

sop(x); //130