How to convert octal to decimal :
----

015 -> Here number '0' describes that It is an octal Literal.

$(\overset{1}{1}\ \ \overset{0}{5})_8 = (\ ?\ )_{10}$

digit X base^{Power} + digit X base^{power}

$1 \times 8^1 + 5 \times 8^0$

8 + 5 = 13

0Xadd -> Here '0X' describes that it is hexadecimal literal.

$(\overset{2}{a}\ \overset{1}{d}\ \overset{0}{d})_{16} = (\ ?\ )_{10}$

a = 10
b = 11
c = 12
d = 13
e = 14
f = 15

$a \times 16^2 + d \times 16^1 + d \times 16^0$

10 X 256 + 13X 16 + 13 X 1

2560 + 208 + 13 = 2781

0b101 -> Here '0b' describes that it is a binary literal

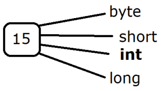
$(\overset{2}{1}\ \overset{1}{0}\ \overset{0}{1})_2 = (\ ?\ )_{10}$

$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

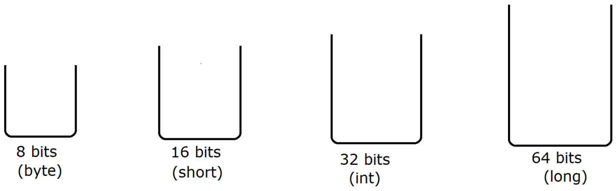
4 + 0 + 1 = 5

---

* By default every integral literal is of type **int** only.



The range of byte data type : -128   to 127
The range of short data type : -32768 t0 32767



* By default any integral literal is of type **int** only, byte and short both are smaller than int, we can assign an int value to byte and short data type but the corrosponding values must be within the range only

```
byte b = 127;  //Valid
byte c = 128;  //Invalid out of the range

short p = 32767;  //Valid
short q = 32768;  //InValid out of the range
```

byte b = (byte) 127; [Here Compiler will convert 127 which is integer value into byte type explicitly]
[This conversion is called Explicit OR Manual type casting] Narrowing

byte c = 128; //Compilation error ,128 is out of the range of byte

short s = (short) 32767; //[Here compiler will convert 32767 into short type explicitly]

int x = 90; //Valid 90 is int type assigning to int type

long y = 1; //Valid, 1 which is int type (32 bits) assiging to long type(64 bits), Automatic Type Casting
[Widening]

long mob = 9812345678; //Invalid the mob number is out of range of integer

long mobile = 9812345678L; //Valid