

HW CLASS:02 — 25/08/23

<https://www.linkedin.com/in/manojofficialmj/>

NUMBER SYSTEM

DETAILED NOTES

① Binary and decimal:

Decimal \rightarrow Total number of difference is 10 (Radix/Base) is called decimal.

EXAMPLE

$(123)_{10} = 100 + 20 + 3$
 $= 1 \times 100 + 2 \times 10 + 3 \times 1$
 $= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$
 $= 123$

HUNDRED PLACE \leftarrow (1)
 TENSE PLACE \leftarrow (2)
 ONCE/UNIT PLACE \leftarrow (3)

Binary \rightarrow Total number of difference is 2 is called Binary.

EXAMPLE

$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
 $= 4 + 0 + 1$
 $= (5)_{10}$

Binary to decimal way-1

Binary to decimal: way-2

2^4	2^3	2^2	2^1	2^0	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal	256	128	64	32	16	8	4	2	1		

Ex ① $(1011)_2 = (11)_{10}$

2^3	2^2	2^1	2^0
1	0	1	1

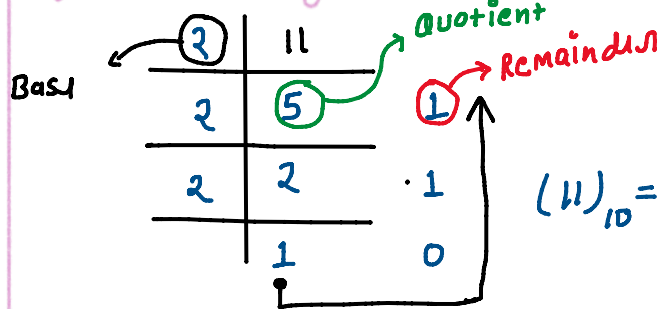
$8 + 0 + 2 + 1 = 11$

Ex 0 $(1011)_2 = (11)_{10}$

2^3	2^2	2^1	2^0
1	0	1	1

$8 + 0 + 2 + 1 = 11$

Decimal to Binary \Rightarrow



$(11)_{10} = (1011)_2$

② 32 bit vs 64 bit processor -

System

Control Panel > System and Security > System

View basic information about your computer

Windows edition

Windows 10

© 2017 Microsoft Corporation. All rights reserved.

System

Manufacturer: LENOVO

Model: 80UE

Processor: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz

Installed memory (RAM): 8.00 GB

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Settings

Home

Find a setting

System

Display

Sound

Notifications & actions

Focus assist

Power & sleep

Battery

Storage

Tablet mode

About

win = 11

Device specifications

Blade Stealth

Device name: Yuen

Processor: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz

Installed RAM: 16.0 GB (15.9 GB usable)

Device ID: 69AB04AE-2F34-40A8-B7F1-A1DD8EA27D30

Product ID: 00000-00000-00000-00000-00000

System type: 64-bit operating system, x64-based processor

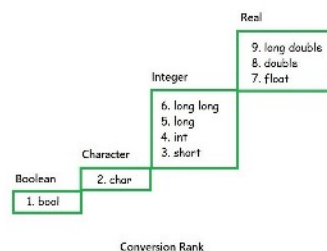
Pen and touch: Touch support with 10 touch points

Rename this PC

PARAMETERS	32 bit OS	64 bit OS
OS SUPPORT	It needs a 32-bit operating system.	It can run both on 32-bit and 64-bit operating system.
COMPATIBLE SYSTEM	It supports windows 7, 8, Vista, XP, and Linux.	Win XP, Professional, Vista, 7, 8, 10, 11, Linux, and Mac OS.
ADDRESSABLE SPACE SIZE	32-bit processors have 4 GB of addressable space.	It has 16 GB of addressable space.
APPLICATIONS SUPPORT	You can not run the applications that are designed for 64-bit.	Both application of 32-bit & 64-bit will support.
MEMORY LIMITS	The 32-bit systems are limited to 3.2 GB and can be used only in 32-bit windows. you won't be able to access your full memory limit of 4 GB.	you can store nearly 17 GB billions RAM. In short, There are no memory limits.
MULTI-TASKING	Not Best	Best pick for it.
SUPPORT TO OLD SYSTEMS	it can be used and supports the old systems that are developed in the early 2000s or late 1990s.	it is more focused on the current system. and don't support an older system like the former.

③ → WHY TYPE CONVERSION (Type casting)

By using casting, data can not be changed but only the data type is changed. Type casting refers to the conversion of one data type to another in program.



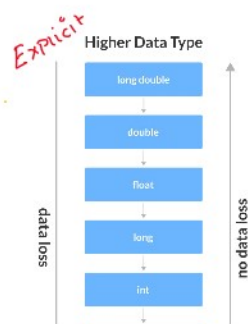
Widening or Automatic Type Conversion

Widening conversion takes place when two data types are automatically (**Implicit**) converted. This happens when:

- The two data types are compatible.
- When we assign a value of a smaller data type to a bigger data type.

```
//implicit conversion example
#include<iostream>
using namespace std;
int main()
{
    int num1 = 10; //integer num1
    char char1 = 'a'; // character char1
    //char1 implicitly converted to int, ASCII value of 'a' is 97
    num1 = num1 + char1; 107
    cout << "num1 = " << num1 << endl;
    // num1 is implicitly converted to double
    double num2 = num1 + 54.8; 161.8
    cout << "num2 = " << num2 << endl;

    return 0;
}
```



Narrowing or Explicit Type Conversion

Narrowing conversion takes place when two data types are not automatically converted. This happens when:

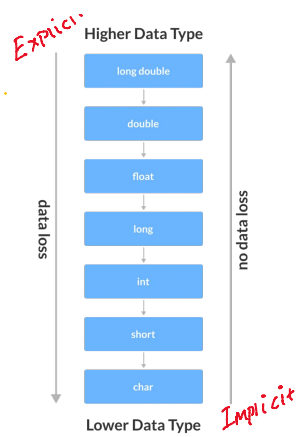
- The two data types are incompatible.
- When we assign a value of a bigger data type to a smaller data type.

```
//explicit conversion example
#include<iostream>
using namespace std;

int main()
{
    double num1 = 1.2;
    // Explicit conversion from double to int
    int num2 = (int)num1 + 1; 2
    cout << "num1 = " << num1 << endl;
    cout << "num2 = " << num2 << endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
    int num1 = 10; //integer num1
    char char1 = 'a'; // character char1
    //char1 implicitly converted to int, ASCII value of 'a' is 97
    num1 = num1 + char1; 107
    cout << "num1 = " << num1 << endl;
    // num1 is implicitly converted to double
    double num2 = num1 + 54.8; 161.8
    cout << "num2 = " << num2 << endl;

    return 0;
}
```



```
#include<iostream>
using namespace std;

int main()
{
    double num1 = 1.2;
    // Explicit conversion from double to int
    int num2 = (int)num1 + 1; 2
    cout << "num1 = " << num1 << endl;
    cout << "num2 = " << num2 << endl;
    return 0;
}
```

HomeWork:

- 32 bit vs 64 bit Architecture
- TypeCasting: Implicit vs Explicit
- Number System: Binary and Decimal