

CS basics.

- 1) Intro
- 2) Features of comp: speed, versatile, Accuracy, reliability, storage
- 3) Booting: starting a comp or comp embedded device
steps 1) switch on power 2) loading OS into comp main memory 3) keep all apps ready to use

* RAM (Random access memory): also the main memory of the comp, volatile, stores programs and data when in use. (eg: 8GB RAM) is volatile in

* Hardware components of comp are:

- 1) Processor
- 2) Main memory
- 3) Secondary memory
- 4) I/P devices
- 5) O/P devices.

* ROM (Read only memory): non-volatile used to store BIOS (also part of main memory)

* Embedded systems: A computer system that is a part of a larger device and which controls how that device operates

* Software consists of both programs & data. Programs are set of instructions for the processor. Data is info needed like character data, image data, audio data etc.

* BIOS: is a firmware (specific class of comp software) that provide low-level control for

a device's specific hardware) used to hardware initialization during booting process and provide run-time services for operating systems and programs runtime service : is a software designed to support the execution of comp programs written in comp language

* types of software:

- 1) system software 2) Application 3) Utility
- a) operating system Eg: MS-word
- b) language processor
- c) Device drivers

* system software: is a software designed to provide a platform for other software. It acts as an interface b/w hardware & user.

a) operating system: stored in ROM loaded after BIOS. It is a system software that manages comp hardware. Manages fns like storing data, memory, returning files etc.

b) language processor. 3 types of languages :

- i) Machine level language
- ii) Assembly level
- iii) High level Eg: English statements.

Language processor converts high level to machine level. It is of 3 types

- a) Assembler: converts Assembly level to MLL
- b) Interpreter: high level to MLL
- c) Compiler : " " " at on go

c) device drivers: is a comp program that operates

a device specific hardware) used to hardware initialization during booting process and provide run-time services for operating systems and programs. runtime service : is a software designed to support the execution of comp progs written in comp language

* types of software:

- 1) System software
- 2) Application
- 3) Utility
- a) Operating system
- b) Language processor
- c) Device drivers

* system software: is a software designed to provide a platform for other software. It acts as an interface b/w hardware & user.

a) operating system: stored in ROM loaded after BIOS. It is a system software that manages comp hardware. Manages f'n like storing data, memory, returning files etc.

b) language processor: 3 types of languages.

- i) Machine level language
- ii) Assembly level
- iii) High level eg: english statements.

language processor converts high level to machine level It is of 3 types

- a) Assembler: converts Assembly level to MLL
- b) Interpreter: high level to MLL
- c) Compiler : " " " " at on go

c) device drivers: is a comp program that operates

or controls a particular type of device that is attached to comp. A driver provides software interface to hardware devices, enabling operating system & other comp programs to access hardware fn. When you attach a new device, you need to install its driver so that the OS knows how to manage it.

* **Applicatⁿ software:** is a program or grp of programs designed for user end. A software that performs a single task type.

- Eg:- Word processing software: Eg Notepad, MS word
- Database Eg: oracle, MS Access
- Spreadsheet Eg: Excel, Apple Numbers
- Multimedia Eg: Fox Real or Media player
- Presentation Eg: MS ppt, Keynotes
- Enterprise Eg: Customer relationship management
- Info worker Eg: Documentation tools
- Educational Eg: Google
- Simulation Eg: Flight & scientific simulators
- Content access Eg: web browsers
- Applicatⁿ suites Eg: Open office, MS office
- Software for engineering & product development (IDE)

* **Utility software:** It is a cross b/w system software & applicatⁿ software Eg i) Antivirus ii) Disk management tools iii) File management tools iv) Compression tools v) Backup tools

* **Operating system:** stored in ROM loaded after BIOS, co-ordinates b/w hardware devices and keeps in a state of readiness, enables computer

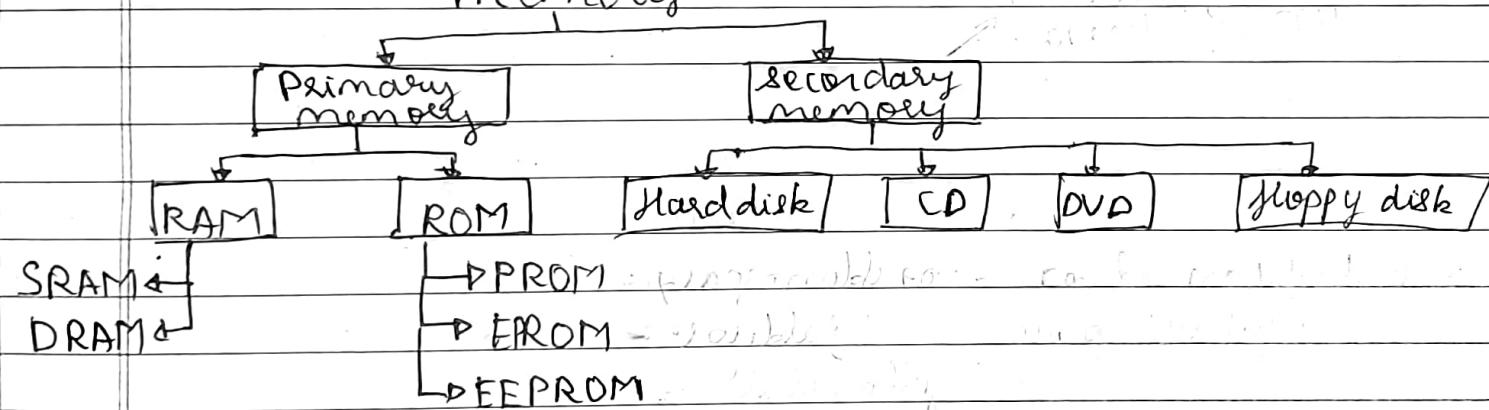
to access network or internet. If managed computer resources like space occupied etc. MS-DOS was the earliest operating system developed by Microsoft. MS-DOS was command line interface (CLI) (means requires command to open softwares) (GUI) - graphical User interface (has icons to open software) Windows was the first GUI-OS developed by Microsoft.

- * Assembler: Is a language processor Assembly level program (source code) MLL (Object code)
- * Interpreter: language processor which converts high level language to MLL line by line takes one line of code, converts and stores it into the object file (is a computer file which contains object code output of an assembler interpreter or compiler)
- * Compiler: language processor which stores the entire program, scans and converts the entire program into MLL from HLL. It requires more space and time than interpreter steps:
 - a) pre-processing : is a program that processes its I/P data to produce O/P that is used as I/P to another program e.g. header files
 - b) lexical analysis : instructions are converted into lexical units like constants, variable symbols etc
 - c) parsing : checking for grammatical errors
 - d) compiling: source code is converted into object code

e) linking: linking to external files or programs

- * Functions of OS: provides service to users collect information and ensures effective utilization of resources.
- * Processor management: is basically jobs scheduling & of two types.
 - a) preemptive scheduling:
 - i) Round robin
 - ii) Response ratio scheduling
 - b) non-preemptive scheduling
 - i) First come first serve
 - ii) shortest job next
 - iii) deadline scheduling

* Memory management:



* primary memory (main memory) used to store data and programs or instructions during computer operat'

* secondary memory: (auxiliary memory) non-volatile & used to store a large amt of data.

* cache memory: serves as buffer b/w CPU & main memory, used to store data and programs that the CPU uses most frequently

PROM - programmable ROM

EPROM - erasable " "

EEPROM - electrically erasable " "

* File management: managing file systems by OS

* Device management: process of implementation, operation and maintenance of a device by OS, acts as interface b/w device & OS

* types of OS: i) GUI - graphical user interface
 ii) time sharing OS iii) Real time OS
 iv) distributed OS

* We can use Linux to build our own OS.

* Number system

Name Base

Binary 2

Octal 8

Decimal 10

Duodecimal 12

Hexadecimal 16

* Decimal: $10^0, 10^{-1}, 10^{-2}$
 are called weights

The value of a decimal number is the sum of the digit times the column weight

$$25 = 2 \times 10^1 + 5 \times 10^0$$

$$25.25 = 11 + \frac{2}{10} + \frac{5}{100}$$

* Binary

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$

$$(11.101)_2 = 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3}$$

$$= 2 + 1 + 0.5 + 0 + 0.125 = 3.625$$

$$\begin{array}{r} 13 \\ \times 2 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 6 - 1 \\ \hline 5 \end{array}$$

$$\begin{array}{r} 3 - 0 \\ \hline 3 \end{array}$$

$$\begin{array}{r} 1 - 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 19 \\ \times 2 \\ \hline 11 \end{array}$$

$$\begin{array}{r} 9 - 1 \\ \hline 8 \end{array}$$

$$\begin{array}{r} 4 - 1 \\ \hline 3 \end{array}$$

$$\begin{array}{r} 2 - 0 \\ \hline 2 \end{array}$$

$$\begin{array}{r} 10011 \\ \hline 10 \end{array}$$

$$(0.825)_10 = ()_2 = 0.1101$$

$$0.825 \times 2 = 1.650 - 1 \text{ (MSB)}$$

$$0.650 \times 2 = 1.300 - 1$$

$$0.300 \times 2 = 0.600 - 0$$

$$0.600 \times 2 = 1.200 - 1 \text{ (LSB)}$$

$$0.200 \times 2 = 0.400 - 0$$

$$0.400 \times 2 = 0.800 - 0$$

Hexadecimal

$$(D5)_{16} = (213)_{10} = 5 \times 16^0 = 5 = 213 \quad \begin{array}{r} 208 \\ 213 \end{array}$$

$$0.82 = \frac{8}{16} + \frac{2}{16^2} = 0.5007$$

Hexa \rightarrow decimal \rightarrow Binary

$$(213)_{10} = ()_{16} \quad \begin{array}{r} 16(213) \\ 16\cancel{13}-5 \end{array} = 135 = D5$$

$$0.625 \times 16 = 10.000 \rightarrow 10 \rightarrow A$$

$$0.625_{10} = (0.A)_H \quad 0.875 \times 16 = 14.000 \rightarrow E$$

$$(27)_{10} = (G)H$$

$$(FF)H = (11111111)_B = (255)_{10}$$

$$(OE FC)H = 0000111011111101$$

$$(110110)_B = 00110110 = (36)H \quad 654 \rightarrow \text{bytes from}$$

Octal

$$(75)8 = (7 \times 8^1 + 5 \times 8^0) = (61)_{10}$$

$$8|68$$

$$8|8-4 = (1048)_8 = (68)_{10}$$

1 - 0

Octal to binary

$$(23)_{10} = 010111 (27)8$$

- * BCD → binary coded decimal → 0 - 9. (also called 8421 code)

$$(17)_{10} = (0001 \ 0111)_{BCD}$$

$$(156)_{10} = (0001 \ 0101 \ 0110)_{BCD}$$

$$(0001 \ 0100)_{BCD} = (14)_{10}$$

- * Gray code two successive values differ in only by 1 bit (0 - 15) (Gray code)

| | |
|---|------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0010 |
| 4 | 0110 |
| 5 | 0111 |
| 6 | 0101 |

(1011) 1110

$$\begin{array}{r} 1011 \\ 1010 \\ \hline 1110 \end{array}$$

- * Hexadecimal addition:

$$4H + 5H = 9H \text{ similar to } 4 + 5 = 9$$

$$5H + 7H = CH \quad " " \quad 5_{10} + 7_{10} = 12_{10}$$

$$\begin{array}{rcl} (94)_{10} & = & 16 \cancel{14} \quad = 514 = (5E)H \\ & & \boxed{5-14} \quad = 5 \times 16^1 + 14 \times 16^0 = 14 + 80 = 94 \end{array}$$

$$(125)_{10} = (6D)H$$

$$\begin{array}{r} 94 \\ 125 \\ \hline 219 \end{array} \quad \begin{array}{r} 5E \\ 7D \\ \hline DB \end{array} \rightarrow E + D = 27 - 16 = (11)B \text{ with carry!}$$

$$\hookrightarrow D \times 16^1 + B \times 16^0 = 13 \times 16 + 11 = 219$$

- program: set of instructions that tell comp what to do.
- Algorithm: step-by-step procedure for calculations. An algorithm is an abstract description of how to perform a task.

$$\begin{aligned}
 & 8_1 + 9_1 = 11_H \Rightarrow 8_{10} + 9_{10} = 17_{10} = 17 - 16 = 1 \text{ with carry } 1 \\
 \hookrightarrow & 12_{10} + 13_{10} = 25_{10} = 16 - 25 = 9 \quad \therefore 9 \text{ with carry } 1 \text{ ie } 19_H \\
 & C_H + D_H = 19_H \\
 F_H + E_H &= (13) \rightarrow D = (1D)_H
 \end{aligned}$$

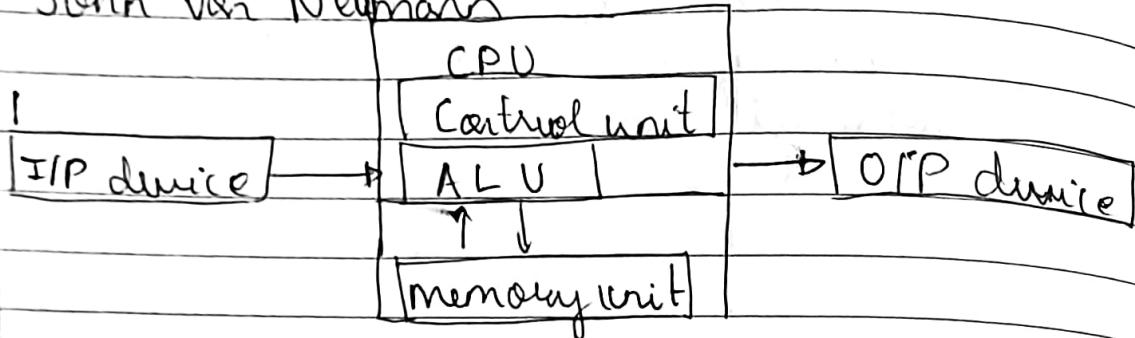
Hexadecimal addition is impf for processors.

$$\begin{array}{r}
 04 : 8 \\
 AS : 9 \\
 \hline
 17A : 1)_H
 \end{array}$$

20/11/21

Computers Hardware

- All modern day computers use Von Neumann architecture, detailed by mathematician John Von Neumann



processing unit \rightarrow ALU & processor register
 control unit \rightarrow instruction register & program counter.

- Von-Neumann bottleneck - share a common bus hence instruction fetch and data operation cannot occur at the same time.

* hardware components.

• system unit.

- mother board (system board)
- CPU
- Memory
- Ports
- Expansion slots

• Secondary storage - CD, Hard disks

• I/P / O/P devices

• Communication devices.

* system unit:

• Mother board - main circuit component

• CPU - ALU + CU

• word size = no. of bits of data a CPU can process.

• system clock: electronic pulse generator today's

Speed = 3.2 GHz.

- Bus lines: electrical data roadways through which data is transmitted between CPU & other components.
- Memory chip
 - RAM - main memory - volatile
 - ROM - non volatile
 - CMOS - (complementary metal-oxide semiconductor) is term usually used to describe the small amt of memory on mother board that stores BIOS setting. Some of these BIOS settings include system time & date, hardware settings
 - Flash - do not require electricity & are non volatile is of type EEPROM
 - Cache - special high speed memory that temporary stores instructions & data the CPU is likely to use frequently. This speeds up processing.
- Expansion slots - slots to plug expansion cards into e.g. of cards = graphic, sound, network cards.
- Ports - sockets
 - serial port transmits one data at a time
 - parallel port transmits 8 bits of data at a time
 - USB - very fast allows multiple devices to be connected to the same port.

* Secondary storage:

used to store data even when power is off, non-volatile

- floppy disk - is removable platter of mylar plastic that a magnetized can hold data of 1.44 MB
- zip disk - 250 MB
- hard disk - same as floppy disk but uses

metal platters upto 100s of GB

- Magnetic tapes

- Optical disk - uses lasers to store data.

 - CD - ROM - compact disk

 - DVD - ROM - digital versatile disc.

* I/P / O/P devices:

- I/P give info to device.

- O/P give O/P

* communication devices: allows comp to send / receive data.

- modem: sends info over phone line

- network cards: sends info over network cable
can be used to hook up comp to LAN

* hyper-threading - Where a single core will present itself as multiple cores to comp OS

* device drivers - designed to talk to devices like scanners, when connecting new devices have to download & install new device drivers

Internet:

- * IP address - internet protocol
- * ISP - internet service provider
- * DHCP - dynamic Host configuration protocol
 - L software that provides IP addresses to your computer.
- * home router supports DHCP & provides IP to every device.
- * DNS (servers) (Domain name system) converts domain names into IP addresses.
- * TCP/IP - transmission control protocol supports sequence number that help data get to its destination
- * protocol - procedure of things need to be done
- * http - hypertext transfer protocol.
- * UDP = user datagram protocol is a communication protocol used across the Internet for especially time-sensitive transmissions such as video playback or DNS lookup
- * Router's purpose is to send data in the direction of destination.
- * traceroute is a program that sends packets to each other router on a path to destination, reporting the time it takes to reach that router
- * MAC address: media access control is a unique identifier assigned to a network interface controller (NIC) for use as a network address in communication within a network segment

This use is most IEEE 802 networking technologies, including Ethernet, Wi-Fi & Bluetooth.

- * Ethernet is a family of wired computer networking technology commonly used in LAN.
- * HTTP = Hypertext transfer protocol is an application-layer protocol for transmitting hypermedia documents, such as HTML.

Web development.

- ★ URL = uniform Resource Locator.
- ★ HTML : Hypertext markup language. - not a programming language.

3/3/23 Intro to CS 5 programming 6:00 (MIT)

- * Computational thinking: is the mental skill to apply concepts, methods, problem-solving techniques and logic reasoning, derived from computing and computer science, to solve problems in all areas, including our daily lives.
 - * Compiles is faster than interpreted language, interpreter (line by line)
 - Compiler takes source code & converts to object code
 - * Python:
 - High level, general purpose & interpreted language

* ~~print~~ ~~input~~ ("Hello" + input("what is name?") + " nice")
O/P -> what is name : psm
Hello psm nice

$c = 3 * 'ab'$ O/P ababab

point (c)

In ASCII, numbers come after alphabets therefore
"a" < 3 → gives false.

• Operator overloading - means giving extended meaning beyond their pre defined operational meaning. Eg '+' is used to add {} join etc also

- tuple is immutable → can not be changed.

deconstruction \rightarrow break code
 Abstraction \rightarrow hide code (black box)

* if ($a > b$):

an

else

if

elif

else

* nested if

if condit:

if condit:

do -

else

do -

else

do -

* $\lfloor \cdot \rfloor \rightarrow$ floor operat^t returns int after rounding a decimal place but not greater than x.

$$\begin{aligned} \text{e.g. } 300.16 &\} = 300.0 \\ 300.72 &\} \\ -23.11 &= -24 \end{aligned}$$

* $\lceil \cdot \rceil =$ smallest int greater than or equal to

$$\begin{aligned} 300.16 &\} = 301.0 \\ 300.72 &\} \end{aligned}$$

$$-23.11 = -23$$

* height $\star 2 \rightarrow$ to the power
 $\text{round}(8/3, 2)$

$$= 2.67$$

* In if elif else \rightarrow only statement matching condition will be executed

* nested if all 3 statements can be executed

Java.

- WORA → write once run anywhere. → Java
- IDE → integrated development environment.
- JDK → java development kit → compiler
- Java → strongly typed
- Java is independent of hardware but not JVM.
 public static void main(String a[])

```

    } system.out.println("Hello World");
}
  
```

2 variables:

assignment int num = 3; → right is assigned to left.
 println → print next line

data types → primitive (eg int, float etc)
 ↳ user defined

Primitive data type

→ int → byte, short, int, long

→ float → double, float

→ character

→ Boolean

int = 4 bytes

long = 8 bytes

short = 2 bytes

byte = 1 byte

float = 4 bytes

double = 8 bytes → default

char = 2 bytes

→ single quote

char c = 'K';

Boolean → true or false

note long lon = 5856L

float num = 5.8f;

boolean b = true;

- int num1 = 0b101
- O/P = 5.
- java allows 10-00-000
- $12e10 = 12 \times 10^{10}$

* explicit type casting \rightarrow casting $b = (\text{byte}) a$

$n++ \rightarrow$ post increment.

$a = ++n$ $n = 8 \quad a = 9$

$a = n++$ $n = 8 \quad a = 8$

- relational operators give boolean values
 - $\hookrightarrow \leq, \geq, <, >$
- logical operators !-NOT, &-AND, ||-OR
 - $s = x > y \text{ if } a < b$
 - $\top \quad \top \quad \top$
- if ($x > y$)
 - {
 - }
 - else if
 - else
- result = $n \% 2 == 0 ? 10 : 20;$

* switch:

```
int n = 2;
```

```
switch (n) {
```

```
case:
```

```
    System.out.println("Monday");
```

```
    break;
```

```
case2:
```

```
    System.out.println("Tue");
```

```
    break;
```

```
default:
```

```
    System.out.println("NO");
```

* loops:-

```

int i = 1;
while (i <= 4)
{
    System.out.println("Hi " + i);
    i++;
}

```

```

do
{
    ...
} while (i <= 4);

```

```
for (int i = 4; i >= 1; i --)
```

* OOP:

Objects have properties & Behaviours.

e.g: class Calculator

```

int a; public return type
public int add (int n1, int n2)
{

```

```
    int s = n1 + n2;
```

```
    return s;
```

```
}
```

```

}
public class Demo
```

```
public static void main (String a[])
{

```

```
    int num1 = 4;
```

```
    int num2 = 5; reference variable

```

```
    Calculator calc = new Calculator(); new object.
```

↳ name of class

int result = calc.add (num1, num2);

System.out.println (result);

}

}

- main is a method

* Method overloading: one method is called with diff return types & diff values

Eg: public class MyClass {

 public static void main (String [] args)

 int x1 = add (3, 2);

 double x2 = add (3.1, 2.2);

}

 System.out.println (x1);

 System.out.println (x2);

}

 public static int add (int a, int b) {

 return a+b;

}

 public static double add (double a, double b) {

 return a+b;

}

int b

?

- Both give O/P