# Computer Systems

computer system / Hardware (physical equipment)
\ Software (collection of program that allows hardware to do work).

Hardware - 5 parts

* Input devices - programs to computer
eg:- Keyboard, pen (or) stylus

* CPU - executing programs such as arithmetic operation

* Primary storage - main memory. Temporary storage of data or program.

* o/p deveices - Display the output -
eg. monitor (or) printer
↓ softcopy ↓ Hardcopy

* Auxillary storage devices - (secondary storage)
↳ Data (or) program stored permentanently.

Software → programs (or) digitalized and automated process.
↓ gui ↓ without human interface

Software / system software / OS
\ system support software
\ system development software
\ Application software / Purpose software
\ application-specific software

System Software:

provides interface between user & hardware but does ^not satisfy user needs

* OS : user interface, file and database access

* system support software: provides sort programs and disk format programs

* system development software :- language translators
↓
program → machine language

Application software

solves user needs to solve their problems.

* General purpose software

↳ more than one application

↳ eg:- database management system, word processor, computer aided design system

* Application - specific software:

↳ software used by accountants.

↳ material requirement planning system.

↳ they ^cannot do generalized task.

Platform dependent & independent in computer system

computer system:-

Operating system + hardware devices

C/C++ program → operated directly by OS
↳ platform independent

Java program → JVM (Platform)
.net - CLR
HTML - Browser
} Platform dependent

Platform dependent:

↳ program cand be developed & compiled only in OS.

eg: c/c++

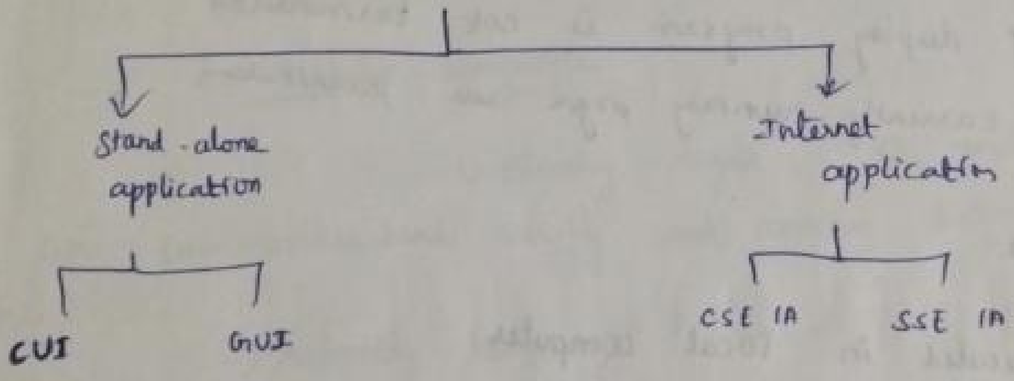
↳ cannot be operated in other platform

Because c/c++ → does not need internet so it is platform dependent.

Platform independent:

↳ program developed and compiled in one OS and can be run in other OS because they are internet based application.

Application of computer system:



STAND ALONE APPLICATION: → eg: Java

↳ executed and installed only in one computer by a single user.

eg: VLC, chrome.

VLC → Installation necessary

Differs from windows & in Android

② types

CUI - character user Interface

GUI - Graphical user Interface

CUI [character user Interface]

↳ Reads the i/p by text based character.

↳ No special window

↳ command window (or) terminal

↳ once written cannot be changed. (sez app is read)

↳ Java, developed by CMD line args.,
(or) scanner (or) by using console.

↳ does not have fonts, graphics. ..

GUI [Graphical user Interface]:-

↳ Reads i/p by separate window

↳ Diff fonts, graphics,

↳ Every internet applications are GUI
all i/p are passed at a tim

↳ o/p display program is not terminated

for new i/p currently running pzgm use panrikalam

eg

Internet application.

↳ Executed in local computer.

↳ multiple computers & multiple user application

2_ types

/ \

CSE IA SSE IA

CSE IA (client side executing internet application,
↳ client sends a request eg: Login page registration

SSE IA (server side executing internet application.
↳ Resides in server system & sends response to client

eg:- JAVA, .NET

Namma kudukura uid ID & pswd check pannathu server system If match with database then server redirect to home page (or) else it show pswd incorrect.

Basics of Programming Language.

program → set of Instructions that are grouped together to perform task.

programming → process of writing programs.

Programming language → medium to express thoughts
Language bt computer & user.

eg:- JAVA, C, C++, python..

Types of prgming language.

* Low level
* Middle level
* High level.

Low level language.

cannot be understand by user.

↳ Machine language

↳ Binary digits 0, and 1 so that cpu can understand easily and process data.

↳ Assembly language:

Set of abbreviations (or) instruction like ADD, MOVE etc., and provides machine language as o/p.

Middle level language.

↳ Intermediate bt. low and high level language

↳ not ~~understand~~ closer to human but can understand

High level language:

* close to humans

* computer cannot understand high level language so it converts to machine level language by using compiler and executes the prgm.

source code: syntax written by programmer.
compiler: source code → machine language at once.
Interpreter: " " " line by line.
compiled code → program generated from S.C by compiler
compilation → process of translating S.C → C.C.
Execution → running of copiled code
Executable code: OS understandable executable prgm.
compile time error: error by compiler due to syntax mistake
spelling (or) wrong usuage of KW.
Runtime error: Error during execution of prgm → due to
logical mistakes

compiler, Intempreter and Assembler:-

↳ They are translators developed by using C, C++
↳ compiler & Interpreter converts [H.L.L → M.L.L]
Assembler [L.L.L to M.L.L]

compiler → whole prgm um check pannum last aah
Interpreter → goes for line by line execution.
If there is error in one line it stops checking
the nxt line.

JAVA programming :-

JAVA → high level language

Easier than c/c++ bcz it has no more inheritance, structures etc.,

SUN microsystems

Platform independent

WORA concept

Features of JAVA

* Simple :-

Study ⇒ 111<sup>y</sup> to c/c++, does not have multiple inheritance, structure, union, template, operator overloading ...

⇒ Inbuilt pointer

Development Predefined libraries (no more codes)

* Secure

⇒ JAVA compiled code doesn't execute directly (Bytecode → verifies byte code)

⇒ Data point of view (accessibility modifiers & incapsulation)

* Robust (int x = 8.2)

* Portable

* Architectural nature (JVM) → any computer that run again

* OOP [class based program → OOP]

* multithread

* High performance.

Other features :

* Byte coded (special instruction set called byte code)

* Interpreted (Byte code → machine language)

* Garbage collector (automatic memory management)

* Open Source [

Applications of JAVA:

Desktop applications

web Servers

Enterprise application. (Bank)

Interoperable application (Facebook)

Mobile application (Android apps)

Gaming application.

Robotics application.

Database connection (oracle)

## JAVA Edition & concepts

(4) → JAVA SE (standard edition) [CUI, GUI]
JAVA EE [web applications, enterprise application]
JAVA ME [micro] → programs in chips → mobile gaming
JAVA FX

JAVA SE ⇒ JDBC, XML, RMI, JNDI

JAVA EE → Servlet, JSP, EJB, webservices, JSF

## JDK, JRE, JVM.

JDK → To develop & run JAVA application

JRE → Run JAVA

JVM → Run prgm line by line

JIT → JAVA byte code fastly (execute)

JVM: [JAVA Virtual Machine]

↳ To run JAVA prgm

↳ Syntax of the source code file is checked

file mistake → compile time error → or b.c → M.L

JAVA runtime environment → JAVA class libraries

JDK - JAVA Development kit

↳ Addition to JRE JDK also contains a no of development tools (compiler, JAVAdoc, JAVA debugger etc.)

JIT:-

Just in Time compiler

↳ Hotspot technology

JVM ⇒ Intreprete + JIT

JRE ⇒ JVM + Library classes

JDK ⇒ JRE + Development tools

## How to run JAVA program :-

In JAVA can be run in online but some don't support it like ideone.com, onlinegbd.com, codepad, codechef.

Running of JAVA in windows

step-1 :- Latest version of JDK should be downloaded from oracle website. .exe (or) .zip file (download)

step-2 .exe file or , for .zip file should be extract it. JDK is installed

step-3 :- To update variables since just by installing software it binary and library files are not available from other directions

so ⇒ Advance system setting → My computer → properties → Advanced system setting

step-4 :- Click on environmental variables.

step 5:- In ~~ur~~ the user variables section or system variables section we must create a path.

step-6. Add, variable name = "path"

variable value is a location where JAVA is installed.

By default JAVA is installed in "c:\program Files\JAVA\jdk\bin" or,

"c:\program Files (x86) \JAVA \jdk \ bin"

click ok → save.

step 7:- To check the installation.

open command prompt & type java -version

JAVA HELLO WORLD PROGRAM

Software required for developing prgm

*Text editor [notepad]

* JDK [compiling & executing] < compiler / JVM

* command prompt [ execution of javac, java & commands]

Other software used → IDE like eclipse.

↳ editing, compilation & execution of prgm

## JAVA Syntax:

For ~~[illegible]~~ Printing "Hello world."

Prgm:

```
public class Main {
    public static void main (string[] args) {
        System.out.println ("Hello World");
    }
}
```

Explanation:

* The name of class should begin with uppercase.

* The class name (Main) should match with the java file name. (so when creating the java file save it with class name with .java extension).

* System.out.Println → This line is for printing the output line.

* Class should contain the main () fnct.

## JAVA print / output:

Println () ⇒ This is used to print the next line statement

print () ⇒ This is not for printing line by line.

The statement should be given inside the double quotes

```
System.out.println (" Hello world ! ")
System.out.println (" I am learning JAVA ")
```

o/p:
```
Hello world
I am learning JAVA.
```

without double quotes ⇒ Error

The numbers can also be printed

For printing numbers double quotes is not necessary

~~Print~~

System. out. println(3)

o/p :- 3

Arithmetic operations can also be performed.

Example:

```
public class Main {
      public static void main ( string[] args) {
         System. out. println (4 +5) ;
      }
}
```

o/p :- 9

JAVA comments:

↳ To explain the JAVA codes

Single - line comments:

~~It starts with // and endwith //~~

It starts with //.

System. out. println ("Hello world") ; // This a comments

↳ Ethu varaikum dhan execute pannanum

JAVA multi-line comment:

Starts with /* and ends with */

Any text bt these will be ignored.

↳ Longer comments.

eg:

```
/* The code below will print the words Hello world
to the screen and it is amazing */
System. out. println ("Hello world");
```

## JAVA Variables

Variables → To store data values

* String → To store the text like "Hello" → double quotes
* int → To store number
* char → To store 'a' → characters (single quotes)
* float → To store ~~Integer~~ values with decimal point.
* boolean → to store values with 2 states True (or) False.

Syntax:

~~type~~ ~~variable~~

type variableName = value;

↳ while declaring the variable type should specified.

String:

```
String name = "Hello";
System.out.println(name);
```

o/p: Hello

int:

```
int myNum = 65;
System.out.println(myNum);
        (or)
int myNum;
myNum = 65;
System.out.println(myNum);
```

o/p: 65

char:

```
char name = 'a';
System.out.println(name);
```

o/p: a

float:

```
float myFloat = 19.5;
System.out.println(myFloat);
```

o/p - 19.5